TONI RODRIGUEZ

Fare musica con Atari

franco muzzio editore

GLI STRUMENTI DELLA MUSICA

Serie diretta da Virginio B. Sala

1 Marco E. Nobili *La chitarra Gibson "Les Paul"*
2 Andrea Centazzo *La batteria*
3 Jerry Cocker *Improvvisazione jazz*
4 Howard Schott *Suonare il clavicembalo*
5 Yehudi Menuhin e William Primrose *Il violino e la viola*
6 Marco E. Nobili *La chitarra elettrica personalizzata*
7 Delton T. Horn *Il sintetizzatore*
8 Jack Brymer *Il clarinetto*
9 Gaspare Scuderi *Beethoven. Le sonate per pianoforte*
10 Giancarlo Chiaramello *Elementi di composizione musicale*
11 Fabio Mariani *Trattato di chitarra jazz - Uno*
12 Fabio Mariani *Trattato di chitarra jazz - Due*
13 Françoise E. Goddard *La voce*
14 Adriano Bassi *Giorgio Gaslini*
15 Lindoro Massimo del Duca *Musica digitale*
16 Marco E. Nobili *Tutto Gibson*
17 Andrea Bornstein *Gli strumenti musicali del Rinascimento*
18 Adriano Bassi *Giacomo Orefice*
19 Ralph Kirkpatrick *L'interpretazione del Clavicembalo ben temperato*
20 Arthur Wills *L'organo*

Toni Rodriguez

Fare musica con Atari

franco muzzio editore

Gli antichi greci consideravano due discipline fondamentali, nella loro educazione: la Mousike e la Gymnopedia. La prima significava la cultura mentale in generale, contrapposta alla seconda, che era la cultura fisica.

Questo libro è stato possibile grazie alla pazienza di: Roberto Casiraghi, Patrizia Crippa, Goffredo Haus, Dina Fantini, Attilio Ferrando, Mauro Torelli, M. Alberta Alberti e Virginio Sala.

Redazione tecnica di Sergio Fardin
Composizione realizzata con Macintosh® e stampante Laserwriter®

Prima edizione: ottobre 1987
ISBN 88-7021-427-3

Indice

Prefazione p. 7

Introduzione p. 9

1 **Per cominciare** p. 13
1.1 Senza problemi con il mouse 1.2 Pochi concetti da imparare per scrivere musica 1.3 Puoi provare a comporre qualcosa 1.4 Il suono e la partitura 1.5 Il timbro e la sua notazione 1.6 Cos'è il CAI, prospettive educazionali per la famiglia degli Atari ST

2 **MIDI e dintorni** p. 75
2.1 Tutto quello che avresti voluto sapere della MIDI... ma non hai mai osato chiedere 2.2 Hardware e specifiche della MIDI 2.3 Il codice MIDI 2.4 Interfacciamento con il computer: architetture e limitazioni 2.5 Problemi e limitazioni del MIDI 2.6 Configurazione ideale di un home studio MIDI 2.7 I sistemi di elaborazione musicale SEM 2.8 La workstation musicale 2.9 Le famiglie di software per la MIDI

3 **Computer music** p. 139
3.1 Perché è così difficile trovare un software per scrivere musica a partire dall'immissione dati via MIDI? 3.2 Un po' di storia dell'informatica musicale per carpire meglio che cosa è realizzabile in un futuro imminente e che cosa tarderà ancora a realizzarsi. 3.3 Lo stato dell'arte 3.4 Le reti di Petri: una rappresentazione formale per la conoscenza musicale

4 **Atari ST e utenti avanzati a livello informatico** p. 171
4.1 Anatomia degli ST 4.2 Pregi e limitazioni del TOS™ 4.3 Altri possibili sistemi operativi per gli ST 4.4 Programmare gli ST e Mega 4.5 Intelligenza Artificiale e ST 4.6 Qualche idea su come sviluppare software per gli ST e qualche utility sulla MIDI 4.7 Una nota finale

Appendice 1 **Software musicale commercializzato (ottobre'87) per gli Atari ST e Mega** p. 197

Appendice 2 **Who's Who** p. 213

Piccolo vocabolario dei termini informatici usati per l'Atari ST e nell'informatica musicale p. 217

Bibliografia p. 230

Prefazione

Se volete realizzare una qualsiasi esperienza musicale (composizione, esecuzione, analisi e stampa di partiture ecc.) l'elaboratore è certamente il compagno ideale: non pone limiti a quello che si può fare; gli unici limiti sono la fantasia umana e la capacità di comunicare le proprie idee in modo formalizzato, cioè la capacità di programmare.
Se non sapete di musica e non sapete di informatica, ma volete occuparvi di informatica musicale, è comunque possibile fruire delle possibilità offerte dagli elaboratori "usando" i programmi che il mercato del software rende disponibili con sempre maggior attenzione anche per il settore musicale.
Questo libro costituisce una introduzione all'informatica musicale con un orientamento specifico agli elaboratori Atari della serie ST e Mega ed è rivolto tanto ai neofiti informatici (già musicisti) quanto ai neofiti musicisti (già informatici), ma anche a chi di informatica musicale ha già un'esperienza più o meno ampia e vuole conoscere le caratteristiche e le possibilità di questa famiglia di elaboratori che si è rivelata di particolare interesse per i musicisti (MIDI incorporata, una buona dose di memoria, costi molto competitivi, grafica evoluta).

La formula espositiva adottata in questo libro è tale da crescere gradualmente dal particolare al generale, dagli aspetti più superficiali (di utenza) ai più profondi (di ricerca e di sviluppo), pur mantenendo un tono discorsivo e piacevole, ricco di aneddoti, e senza peraltro perdere niente in termini di "pulizia" concettuale.
Non vi resta che leggerlo.

Goffredo Haus
Direttore Scientifico del Laboratorio di Informatica
Musicale dell'Università degli Studi di Milano

Introduzione

Fa caldo, davvero fa troppo caldo a Milano questa estate, ma a pensarci bene in inverno fa troppo freddo... però forse non sono le questioni climatiche a suscitare il vostro interesse.
Sono nella "stanza delle macchine" davanti a un Atari 1040ST collegato a una tastiera e a qualche altro "aggeggio" MIDI, circondato da varie pile di dischetti, libri, fotocopie d'articoli..., mi sento felice, so che posso raccontarvi delle cose interessanti e già ho cominciato a battere sui tasti della mia alfanumerica (tastiera). Dunque comincio a scrivervi.
Ma a chi sto scrivendo? Com'è l' immagine del mio interlocutore? Quando scrivo un articolo scientifico, o divulgativo, non ho mai questo problema, l'articolo ha il suo pubblico specifico, sai fin dall'inizio quali sono i bisogni d'informazione del tuo destinatario, e se sei esperto dell'argomento sicuramente coprirai il suo interesse.
Invece scrivere un libro che parla della musica (grandissima parola!) e di un elaboratore col marchio Atari, è difficile.
Atari è stato fino ad oggi il marchio leader nel mercato degli elaboratori per giocare o dei videogiochi (come comunemente sono chiamati queti tipi di

computer), le sue macchine erano macchine solide, estremamente affidabili, però molto orientate verso questo settore e dunque normalmente poco flessibili. Invece questi Atari ST (512, 1040 e successivi) sono proprio macchine super-flessibili ed estremamente potenti, un approccio molto diverso rispetto a quello delle macchine precedenti.
Logicamente questo lo so io (e molta altra gente), però può darsi che qualcuno ancora pensi di trovarsi di fronte a una macchina-videogioco... In realtà non importa mica tanto: le macchine per videogiochi sono fortemente condizionate dal fatto di essere molto semplici da usare, e questo non è un peccato, anzi!
La tendenza oggi (tra l'altro assolutamente giusta e utile) dell'informatica applicativa è di fornire all'utente uno strumento per lavorare e risolvere i suoi bisogni, in maniera potente o complessiva, però di semplice uso.
Questa tendenza chiamata "*end user*", cioè orientata all'utente finale, ha popolarizzato l'uso del "mouse" per gestire con i suoi "clic" le finestre interattive (i dialoghi con l'elaboratore) e i comandi attraverso la presentazione di "menù", e anche "vestendo" le applicazioni con grafiche ogni volta più evolute, e soprattutto molto evocative.
Le macchine Atari ST sono proprio fatte secondo questa linea, hanno capacità grafiche molto potenti e anche capacità di calcolo e memoria altissime, il tutto al prezzo più contenuto del mercato.
Penso che questo rapporto qualità-possibilità-prezzo sia la miglior offerta in questo momento nel mercato dell'informatica musicale: è probabilmente il miglior argomento per spiegare il successo di queste macchine in questo settore.

Questa prima considerazione mi porta a pensare che sicuramente una parte dei lettori possa tranquillamente non sapere nulla né di elaboratori né di musica. Ma l'essere interessati a questo tipo di macchine, visto che sono facili da usare, può servire all'utente, ad esempio, per imparare la musica.
Dunque dedicherò la prima parte del libro a questo interlocutore (tra l'altro il più numeroso a livello statistico) e lo farò entrando in materia partendo dal livello "zero, zero": niente informatica, niente musica.
Considerando il fatto che le macchine Atari ST hanno tutte l'interfaccia MIDI nella loro configurazione standard (e questo significa un importante risparmio economico e di installazione a chi già usa strumenti MIDI, ed è senz'altro un aumento importante di precisione e velocità nella gestione della porta MIDI), queste macchine sono sempre più usate dai musicisti informatizzati.
I "musicisti informatizzati via MIDI" costituiscono una razza tutta speciale. Io in un certo modo sono uno di loro (e raramente si può trovare qualche

informatico musicale assolutamente libero da questa malattia infantile).
Dunque dedicherò il secondo capitolo a loro, che formano l'esercito più sofferente, ma anche il più divertente e numeroso dell'ambiente informatico-musicale.
All'interno di questa categoria, la sensazione generale è la confusione, confusione provocata dall'acquisizione di informazione non normalizzata, attraverso il martellamento della pubblicità (diretta o indiretta), chiaccherate con gli amici, stoici acquisti di "prodotti bidonari", mancanza d'assistenza e soprattutto mancanza di una visione generale del problema, che potrebbe far luce, ordine e logica nel "puzzle" delle informazioni.
Per capire meglio il secondo capitolo converrebbe pulire la mente; il mio consiglio sarebbe rileggere soprattutto i concetti musicali che si trovano nel primo capitolo, in particolar modo per coloro che provengono dalla musica "a orecchio".

Ci sono molti campi legati alla musica, dove l'elaboratore è già presente da molto tempo; per fare un esempio immediato il campo dell'editoria: tutti conoscono l'uso dell'elaboratore per i processi di composizione di libri, giornali e altri documenti (e anche per la loro stampa). Per la composizione di partiture musicali esistono già da tempo prodotti software capaci di raggiungere con maggiore o minore efficienza lo stesso scopo; questo settore dell'informatica musicale viene chiamato *editoria musicale elettronica.*
Nello specifico campo musicale, già da tempo l'elaboratore si utilizza per funzioni meno conosciute. Lo studio dell'impiego degli elaboratori a fini musicali costituisce il "corpus" dell'*informatica musicale*: è una disciplina relativamente giovane, ma che ogni giorno di più coinvolge informatici, studiosi della musica e musicisti in generale.
Nel terzo capitolo farò un po' di storia dell'abbinamento tra l'informatica e la musica: dell'informatica musicale come disciplina.
Parlando in generale, penso che questo capitolo si possa rivolgere alle persone, musicisti e non, più sensibilizzate verso l'informatica musicale (IM) o semplicemente curiose di sapere le strade che sta percorrendo oggi l'informatica musicale o *computer music* (nome con cui è conosciuta questa disciplina negli USA) a livello sia europeo che mondiale.
Logicamente il terzo capitolo riprende discorsi già iniziati nei capitoli precedenti e li riconduce nell'ambito generale dell'informatica musicale.

Il quarto capitolo (che chiude il libro a livello di esposizione di temi vincolati alla coppia "musica-Atari ST") è una guida all'uso degli Atari ST per gli utenti

più avanzati a livello informatico; tutte le informazioni possono essere complementate da altre letture e soprattutto con altre esperienze nel campo dell'implementazione software, comunque questo capitolo può servire anche come provocazione allo studio dell'informatica, in particolar modo per i più interessati al nuovo e che sono riusciti a individuare (questo nuovo) attraverso la lettura dei capitoli precedenti (magari!).

I libri possonno avere vita lunga o corta, dipende da ciò di cui parlano e da come sono concepiti. Un libro che parla di una serie di macchine normalmente coprirà l'interesse dei potenziali utenti di queste macchine e avrà vita pari alla vita di quelle macchine. Per questo ho pensato di separare il testo fondamentale (i quattro capitoli) dalla recensione dei prodotti commerciali (software principalmente), che hanno una vita diversa dal libro e dalle macchine.
La prima parte delle Appendici è stata pensata per far da guida ai lettori sui prodotti che già si trovano nel mercato, ma anche per quelli in sviluppo al momento della stesura di questo testo; successive edizioni potranno aggiornare queste appendici.

Il maggior problema dell'uso degli elaboratori è senza dubbio il doversi avvicinare a una terminologia nuova, o perlomeno estremamente specifica; per questo la seconda parte delle Appendici è dedicata a un dizionario di semplice consultazione nel quale sono elencati tutti i termini strettamente informatici corredati di traduzione (quelli che provengono dall'inglese) e spiegazione.

La bibliografia elenca referenziatamente una serie di testi importanti sui temi qui accennati, il che può permettere ai lettori più interessati di continuare a informarsi sui problemi che li preoccupano, e pertanto non è concepita come una bibliografia in senso stretto, ma come una "guida ad altre letture".

Who is Who? vuole fornire l'indicazione delle società produttrici di software, di strumenti musicali e via dicendo, delle società che li commercializzano nelle diverse aree geografiche, degli Enti che svolgono attività di ricerca in informatica musicale.

Il caldo intanto continua...

Milano, luglio - agosto 1987

1

Per cominciare

La musica rivela i principi (Noumena), piuttosto che le mere apparenze (Phenomena) della Natura.

Platone

Siamo di fronte a una macchina che assomiglia molto a una macchina da scrivere, con uno schermo simile a quello della TV: questa macchina è un Atari ST. La prima cosa che devi fare è leggere attentamente le istruzioni di collegamento e avviamento fornite dal costruttore, normalmente le troverai nella sezione "*Getting Started*" del manuale.
Una volta collegato correttamente, il computer dovrebbe funzionare; se non funziona significa che è mal collegato; non pensare che sia rotto, fai i dovuti accertamenti ed accendilo.

Gli Atari ST hanno un sistema operativo chiamato TOS™ che risiede permanentemente dentro il computer, concretamente nella ROM (*Read Only Memory*), pertanto non è necessario caricarlo ogni volta.
Gli Atari ST per mostrarti la loro prima schermata hanno bisogno solo che tu prema l'interruttore d'accensione.
Gli Atari ST si presentano a te usando come ambiente di lavoro la metafora di una "scrivania" (*desktop*); gli accessori di questa scrivania (*desktop accessory*) non si trovano nel sistema operativo, e vanno quindi caricati.

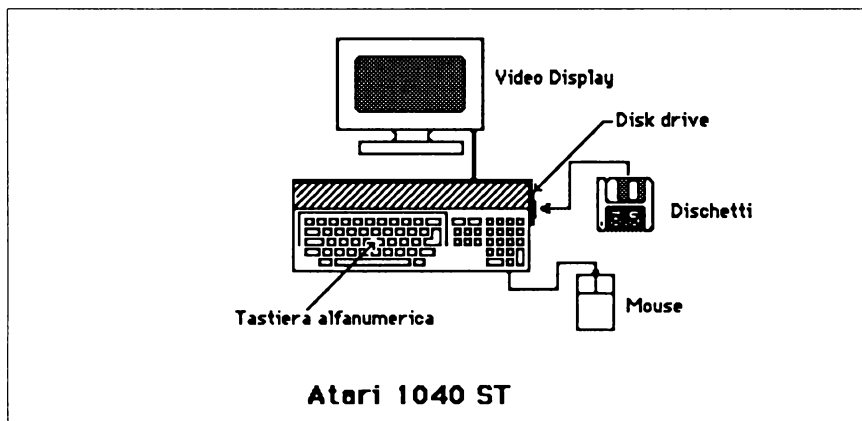



Fig. 1.1

Questi accessori sono i seguenti file: CONTROL.ACC ed EMULATOR.ACC. Il primo disco che devi caricare per completare l'avviamento dell'ambiente di lavoro (*boot*) è il disco che contiene questi due file, che normalmente si chiama *desktop disk*, o, come altri lo chiamano, *boot disk*.
Adesso funziona! È il momento di provare un programma.
Il programma risiede in un supporto e viene caricato nella memoria chiamata RAM (*Random Access Memory*) del computer per poter essere eseguito .
Il processo, quando il supporto che contiene il programma è un dischetto (come nel nostro caso), consiste logicamente nel caricare prima il dischetto.
Caricare un dischetto è semplice: devi immettere il dischetto dentro il driver, cioè nella bocca laterale dove entrano i dischetti (oppure dentro la bocca del floppy disk opzionale — la differenza di impiego puoi trovarla nel manuale di uso, viene trattata nella sezione "A Second Disk Drive ")
Una volta che il dischetto è dentro, devi usare il "mouse"...

1.1. **Senza problemi con il mouse**

Lo schermo iniziale in un Atari ST è quello che si vede in figura 1.2.
Per aprire il dischetto che contiene il programma che ti interessa caricare in memoria, è necessario semplicemente usare il mouse.
Muovendo il mouse, muovi la freccetta sullo schermo: posizionala sopra

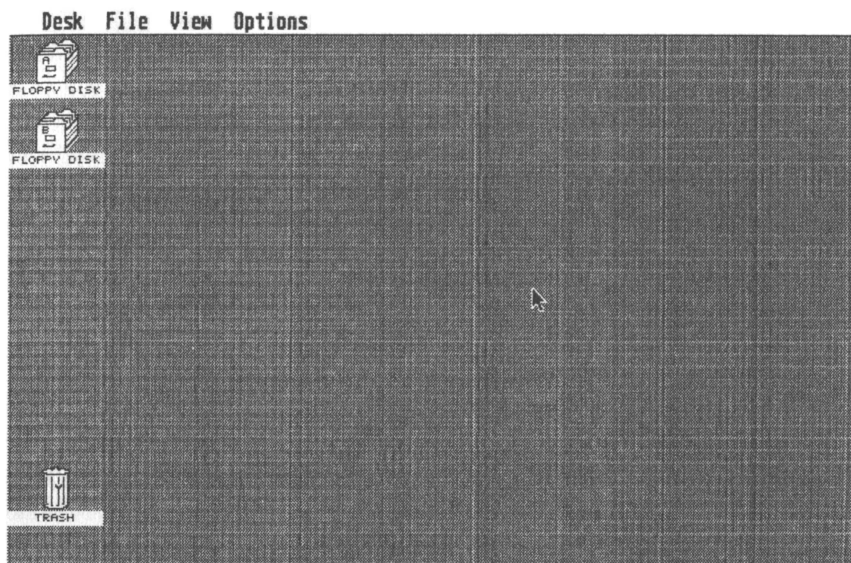



Fig. 1.2 Lo schermo iniziale.

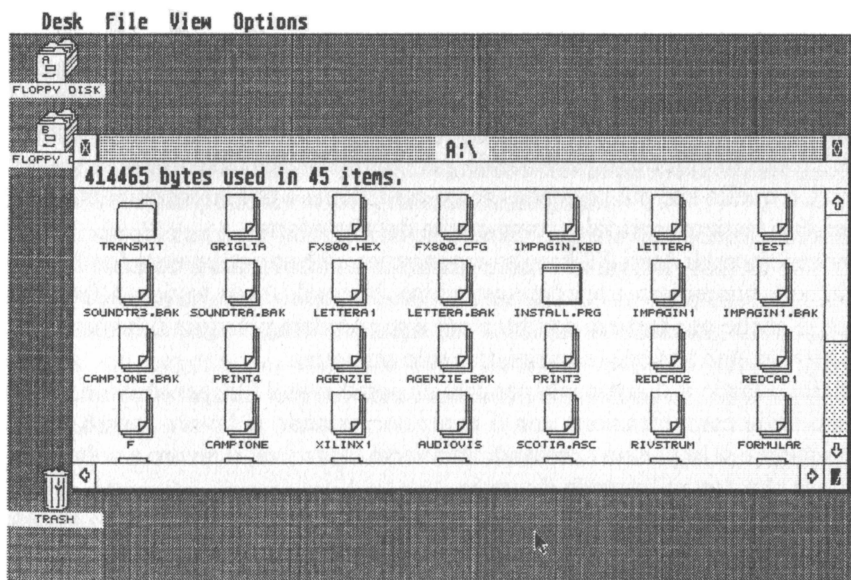



Fig. 1.3 Finestra generale del contenuto di un dischetto.

l'icona del "floppy disk A" ; devi solo fare un doppio clic veloce nella parte sinistra del mouse, subito si aprirà una finestra nel centro dello schermo dove troverai le diverse icone che rappresentano il contenuto del dischetto.
Ciò che hai fatto e che rappresenta un paio di azioni (posizionare e premere il pulsante — per fare i clic —), ti ha risparmiato una grande quantità di operazioni rispetto ad altri sistemi meno "orientati all'utente finale".
A partire da questo momento, ogni volta che vorrai fare qualcosa, dovrai consultare i "menu-bar", posizionarti con la freccetta sopra il comando che vuoi attivare e dare un ordine con i clic del mouse.
Invece quando il computer ha bisogno di dirti qualcosa, o meglio ha bisogno di una tua decisione interattiva, potrai dialogare attraverso una finestra in cui troverai diverse "opzioni" racchiuse in diversi rettangolini che ti permetteranno di rispondere, in diverse maniere, a seconda di dove posizioni la freccetta del mouse e di dove fai il clic.
I percorsi e le scelte del tuo lavoro li potrai costruire attraverso la gestione delle icone (che piano piano capirai corrispondere a cose diverse, a seconda della loro forma grafica-evocativa e del loro suffisso — pippo.PRG diverso da pippo.ACC e anche da pippo.RSC) che risponderanno ai clic del tuo mouse.

In altri elaboratori l'unico mezzo di comunicazione uomo-macchina è la tastiera (*keyboard* in inglese), chiamata *alfanumerica* (perché ha lettere e numeri), da non confonderla con altri tipi di tastiera come la *numerica* (quella dei piccoli calcolatori), la *funzionale* (tastiera che invece di lettere o numeri ha funzioni, normalmente per applicazioni concrete, come ad esempio le tastiere per controllo di processi industriali, per controllo di un ascensore, per aerei ecc.), o quella che più potrebbe creare confusione a noi informatici musicali e cioè la tastiera *musicale* (come quella del pianoforte).
La tastiera negli Atari ST è molto ergonomica ed è uno strumento fondamentale nella comunicazione uomo-macchina. Non solo è una tastiera alfanumerica, è anche una tastiera con funzioni e per rendere più agile il lavoro con i numeri ha una sezione (alla sinistra) solo numerica.
Userai la tastiera sicuramente per dare un nome ai tuoi file, per scrivere, come aiuto nella comunicazione con il computer; quando si lavora spesso con il computer e si imparano i comandi attraverso la tastiera, il lavoro a volte è più veloce che non utilizzando il mouse.
In qualche caso il mouse non sempre ti servirà per comandare il tuo Atari ST, ci sono molti programmi applicativi nei quali il mouse non è totalmente implementato e devi quindi rispondere con tasti funzionali o comandi via tastiera; non ti devi preoccupare: in ogni caso nello schermo troverai indica-

zioni del tasto con il quale devi operare; questo è stato sino ad oggi il modo tradizionale di lavoro con gli elaboratori e per molte applicazioni è ancora oggi molto pratico.
Sicuramente esiste già una generazione di utenti degli elaboratori che ha capito che la tastiera alfanumerica non rappresenta l'unico cordone ombelicale tra loro e la macchina e questo, concettualmente, è positivo e importante.
La comunicazione uomo-macchina , in un futuro prossimo, si orienta verso la strada dell'essere il più "naturale" possibile (umana direi io).

Avevamo lasciato l'Atari ST con la finestra del dischetto aperta; entriamo con la freccetta dentro la finestra e identifichiamo il programma che ci interessa (lo identificheremo facilmente perché ha un tipo di icona caratteristico e porterà il suffisso .PRG), posizioniamo la freccetta sopra e facciamo un clic con il mouse, l'icona cambierà colore e ciò significa che ha accettato la nostra selezione; per caricarlo dobbiamo solo fare un doppio clic veloce nella parte sinistra del mouse (clic,clic!!).
Lo schermo annuncia il passaggio al programma e il cambio d'immagine dalla freccetta all'ape ci indica che il programma si sta caricando... fra pochi secondi comparirà il programma con i suoi "menu-bar" e potremo cominciare a lavorare!

Lo scopo di questo testo non è mostrarti come si usa un Atari ST; quel tipo di informazione è meglio trovarla nello studio tranquillo del manuale. Io ho voluto, seguendo il percorso logico di un ideale primo caricamento di un qualsiasi programma (ovvero partendo da zero), fermarmi a chiarificare quelle piccole cose che potrebbero poi generare confusioni.
Il primo problema è che compaiono parole nuove come file, desktop, mouse, clic; il secondo è che queste parole si presentano in inglese, così ho pensato che il miglior modo per aiutarti è quello di fare un piccolo dizionario; a partire da questo momento qualsiasi parola che esprima un concetto, un'unità di misura, un oggetto, un comando ecc., insomma qualsiasi parola specificamente informatica, la puoi consultare nel vocabolario che si trova nelle Appendici.

Con la lettura del libro progredirai nella conoscenza dei termini informatici. Per adesso hai bisogno solo di saper caricare un programma e usare il mouse.

1.2 **Pochi concetti da imparare per scrivere musica**

Fare musica ... Si può fare musica senza saperla scrivere?
Questa è stata una vecchia discussione, tra l'altro abbastanza inutile, perché si può parlare italiano senza saperlo leggere e scrivere?
Sì, però difficilmente una persona che non sa leggere e scrivere potrà usufruire dei vantaggi culturali derivanti dalla lettura e dalla scrittura e non va dimenticata la funzione della scrittura che è una memoria esterna alla memoria del cervello e comune a tutti gli esseri umani.
Raggiungere un livello di cultura elevato può diventare difficile senza conoscere il patrimonio culturale anteriore e la conoscenza di questo è impossibile senza la lettura.
Questo succede anche con la musica.
Però il problema con gli elaboratori non é la possibilità, ma la necessità . Per lavorare con gli elaboratori, per fare musica con loro, è molto meglio sapere scrivere musica.
Gli elaboratori oggi "capiscono" benissimo la struttura della musica perché esiste una gran quantità di software che in modo diretto o indiretto fa riferimento ai simboli musicali provenienti dalla notazione musicale tradizionale.
La *notazione* è una codifica, storicamente nota, molto ben strutturata, che esprime con chiarezza molti degli eventi musicali; è la codifica nella quale si trova praticamente la maggior parte del patrimonio culturale della musica occidentale "temperata".
È certo che non copre musiche non occidentali, ed è anche certo che si potrebbe prescindere da essa per fare musica (e questo cercheremo di giustificarlo più avanti), però data la sua fondamentale importanza storica e la sua estesissima conoscenza (nel mondo oggi questa notazione si studia in tutti i paesi), diventa inevitabile il suo studio (anche minimo) .

1.2.1. *I concetti basilari*

La misura del tempo:
per convenzione si decide che l'unità di tempo di un determinato brano è *la battuta* .

La notazione:
l'evento musicale minimo è *la nota.*

Essa ha due attributi:

- *altezza* (fisicamente uguale alla frequenza dell'armonico principale della sua composizione di armonici, misurabile in hertz)
- *durata* (fisicamente uguale alla quantità di tempo di emissione dell'evento sonoro — escludendo la riverberazione — misurabile in unità di tempo: microsecondi, secondi, minuti ecc.)

Questi attributi (che coincidono con due delle quattro caratteristiche del suono) non vengono strutturati rispetto a un'organizzazione strettamente vincolata alla misura fisica, ma a una maniera di rappresentazione grafica mediante diversi simboli evocativi e correlati con la realtà fisica del suono.

La codifica dell'altezza

Le note rispetto alla loro altezza sono dodici:

1 = do
2 = do#
3 = re
4 = re#
5 = mi
6 = fa
7 = fa#
8 = sol
9 = sol#
10 = la
11 = la#
12 = si

o anche :

do, re*b*, re, mi*b*, mi, fa, sol*b*, sol, la*b*, la, si*b*, si

perché:

```
do# = re b
re# = mib
```

fa# = sol*b*
sol# = la*b*
la# = si*b*

E questi dodici suoni diversi si ripetono in tutta l'estensione udibile ciclicamente, rispettando sempre il fatto che la differenza tra una nota e la sua omonima successiva è un'*ottava* (fisicamente l'ottava coincide con il rapporto fisico 2 × Frequenza della nota omonima anteriore, in questo modo tutte le note omonime sono armonici fisici di loro stesse).

La distanza fisica tra una nota e la successiva corrisponde per convenzione (convenzione chiamata *temperamento*) a:

$$f_{SUCC} = (f_B) \times (\sqrt[12]{2})$$

dove: f_{SUCC} è la frequenza della nota successiva

e f_B è la frequenza della nota precedente

Questo rapporto significa che lo spazio tra una nota e la sua omonima successiva (l'ottava), si è frazionato in dodici parti uguali, chiamate anche *semitoni*. Oggi la nomenclatura sopraelencata, detta italiana o latina, è usata solo nei paesi di cultura latina (Italia, Francia, Spagna, Portogallo, America Latina), mentre il resto del mondo usa la nomenclatura anglosassone, che usa le lettere dell'alfabeto:

C = do
D = re
E = mi
F = fa
G = sol
A = la
B = si

Il motivo per cui ci sono dodici eventi diversi e solo otto nomi-lettere risale a ragioni storiche, perché in origine la codifica era nata "a solo otto" (però non la musica che voleva rappresentare). Per approfondire questi problemi e altri successivi è meglio leggere i testi indicati nella Bibliografia a fine volume.

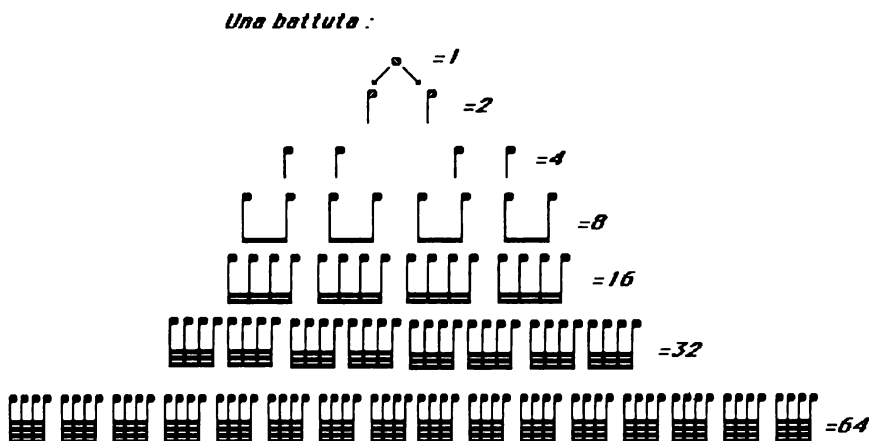
La codifica di durata

Considerando che l'unità di tempo astratta della musica è la battuta, la durata delle note viene riferita sempre alla battuta e corrisponde alla seguente tabella. Per ogni tipo di durata esiste un simbolo diverso:

= *semibreve*
= *minima*
= *semiminima*
= *croma*
= *semicroma*
= *biscroma*
= *semibiscroma*

Nella terminologia anglosassone le note ricevono il loro nome dalle frazioni, rispetto alla battuta da quattro quarti intesa come "battuta ideale":

= *Whole*
= *1/2*
= *1/4*
= *1/8*
= *1/16*
= *1/32*
= *1/64*

La proporzionalità di valori di queste figure può essere rappresentata da un rappresentazione cosiddetta ad albero, che è assai conosciuta:

Nella nomenclatura anglosassone la battuta riceve il nome di *measure*.

La *notazione* è una codifica grafica, che sfrutta come riferimento la posizione delle note rispetto a un asse ampliato e a una serie di righe orizzontali.
Quel *campo* di righe, per convenzione, si estende esplicitamente solo a cinque (quelle che sempre si vedono), per questo lo si chiama *pentagramma;* più popolare ancora la denominazione generica di *rigo*:

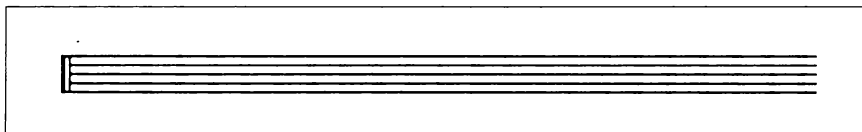



Fig. 1.4 Il rigo (o pentagramma).

Quando hai bisogno di scrivere note al di fuori del rigo (le cinque righe esplicite), si aggiungono dei "pezzetti" di riga (*linee supplementari*) là dove è necessario nell'asta della nota.

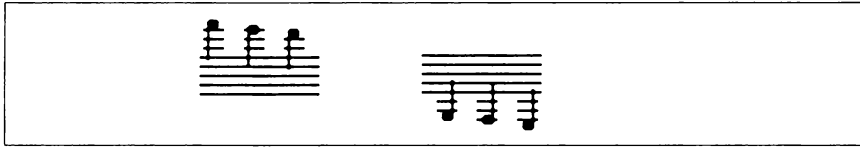



Fig. 1.5 Uso delle linee supplementari.

Per convenzione il rigo corrisponde a un punto di riferimento chiamato *chiave*; poiché le righe permesse sono poche (solo cinque) per coprire tutta la gamma dei suoni possibili (e farlo con linee supplementari rende più difficile una lettura veloce), è comodo spostare il punto di riferimento
È per questo che esistono diverse “chiavi”.

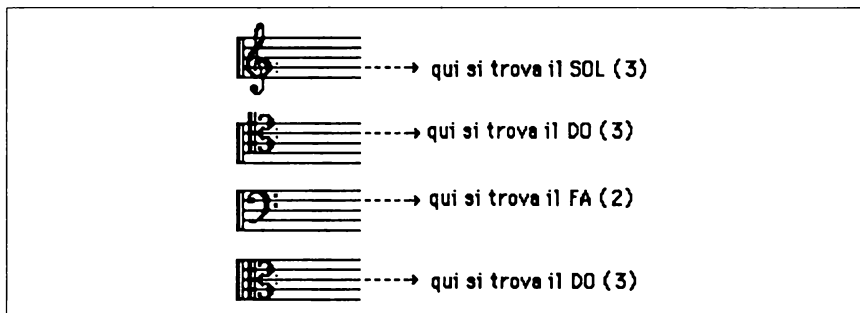



Fig. 1.6 Le chiavi come punto di riferimento.

Di tutte le chiavi che ci sono, storicamente parlando, in pratica ne sono sopravissute solo quattro:

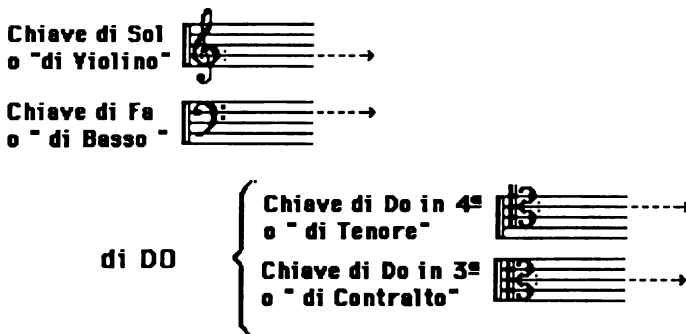
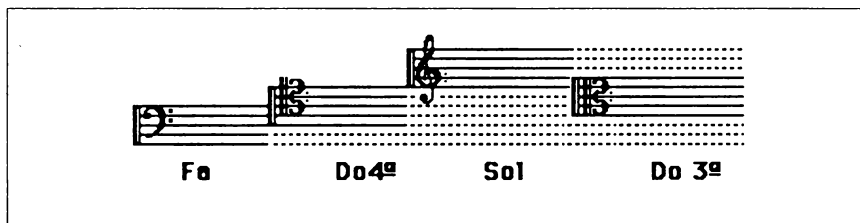



Fig. 1.7 Equivalenza tra le chiavi.

Queste chiavi (che coprono l'estensione della voce umana) hanno un'equivalenza, nel senso che rappresentano una stessa nota in maniera diversa. Questa equivalenza è evidenziata nel grafico di figura 1.7.
Una volta visto questo, vediamo come si inseriscono le note nel rigo:

Nell'esempio la prima "scala" si estende dal do3 (C3) — centrale nella tastiera del pianoforte — al do4 (C4).
Invece la terza scala corrisponde alla quarta ottava e copre dal do4 (C4) al do5 (C5).
La logica di ordine nel rigo è in base a sette note diverse come nella nomenclatura e non a dodici come sono gli eventi reali possibili.
Per approfondire l'uso sistematico delle diverse chiavi, la figura 1.8 mostra alcuni esempi che illustrano tutte le equivalenze di scrittura.
Tra tutte le combinazioni possibili tra le chiavi, la pratica porta a scrivere la chiave di sol e la chiave di fa abbinate; questa scelta è logica perché così si può coprire la massima estensione. Questo modo di scrivere è usato generalmente per il pianoforte e le tastiere, perché sono gli strumenti di maggior estensione

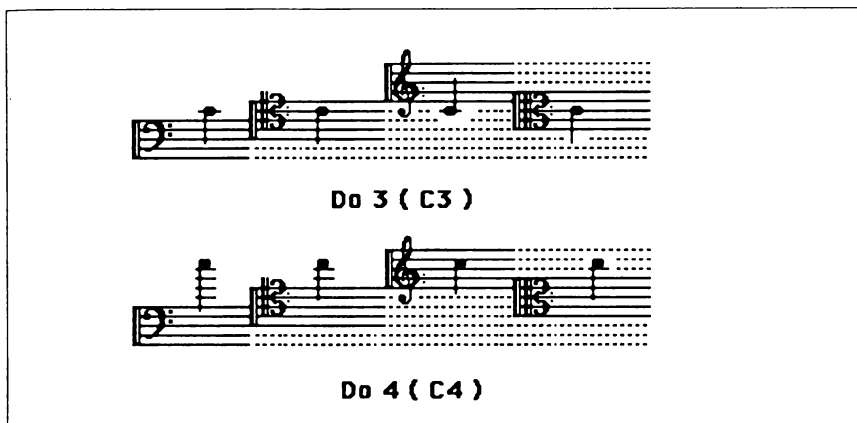



Fig. 1.8 Equivalenza tra scritture della stessa nota in chiavi diverse.

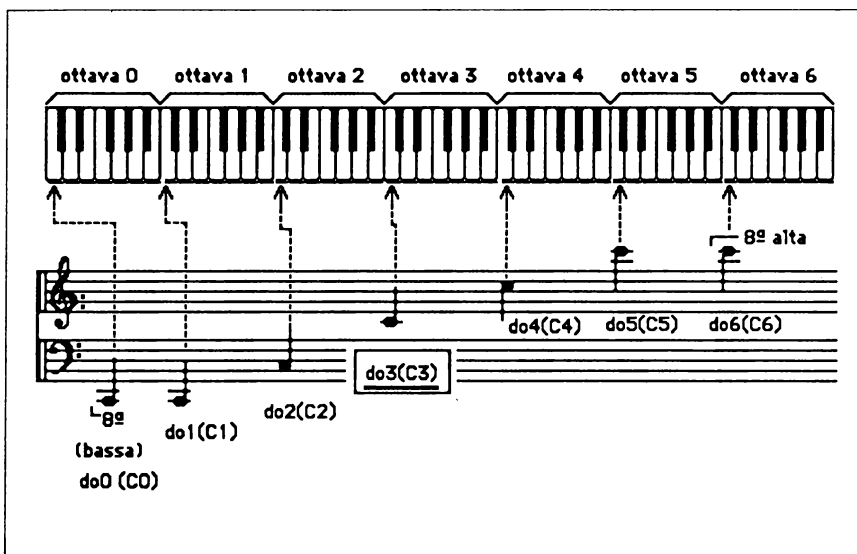



Fig. 1.9 Relazione tra la posizione delle note sulla tastiera e nella notazione sul rigo musicale.

Nell'esempio della figura 1.9, vediamo chiaramente che l'abbinamento tra le due chiavi copre una grande quantità delle note possibili.
All'estremo destro e sinistro, sempre della figura precedente, si trova un'altra maniera di risparmiare linee supplementari. Questo è indicato dal trattino posto sopra la nota con l'indicazione 8ª: ciò significa che quella nota è una ottava superiore. Allo stesso modo, qundo il trattino è posto sotto con l'indicazione 8ª, significa che quella nota è una ottava inferiore.
Nella figura 1.10 vediamo, usando l'esempio della tastiera, la corrispondenza tra tasti e note.

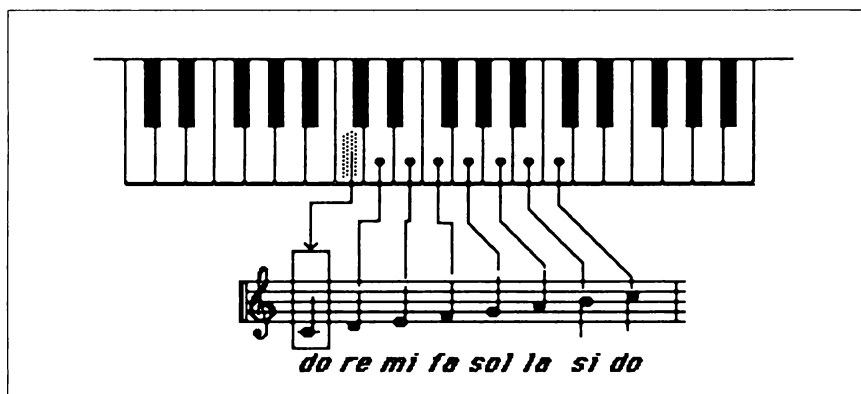



Fig. 1.10 Rapporto tra tasti e note.

1.3. **Puoi provare a comporre qualcosa**

L'unico punto che resta oscuro è senza dubbio dovuto al fatto che sono dodici gli eventi diversi mentre solo sette è il numero di nomi-lettere diversi.
Usando l'esempio della tastiera:

- i tasti bianchi diversi sono sette e corrispondono ai sette nomi-lettere delle note;
- i tasti neri sono cinque e corrispondono a eventi diversi dalle sette note che hanno nome, o lettera, nella nomenclatura anglosassone.

Come facciamo per scriverli?
Prima di finire la spiegazione di tutte le norme di scrittura senza un perché che

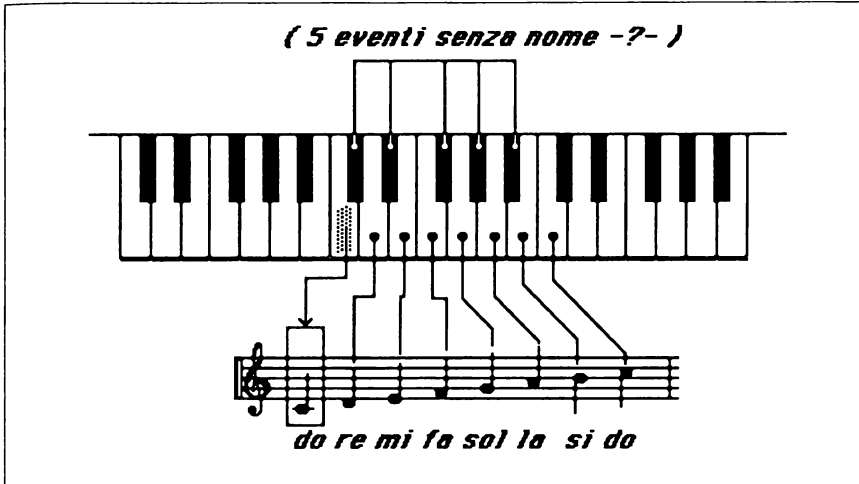



Fig. 1.11 Cinque eventi senza nome.

non sia "perché sì!", preferisco (e molto furbamente, come potrai capire) farti usare i pochi concetti di scrittura che già hai imparato *per fare musiche nuove*, naturalmente con l'ausilio dell'Atari ST e di qualsiasi piccolo programma di scrittura o "sequencer", giocando con l'immissione di dati "step by step", o anche da tastiera, se sei capace di suonare quello che pensi.
Prima di tutto vediamo un piccolo giallo storico. I misteri da risolvere sono due:

- il primo: "Perché ci sono cinque eventi — quelli che corrispondono ai tasti neri della tastiera musicale — che non hanno nome proprio?".
- il secondo: "Perché questi stessi eventi possono chiamarsi in due modi diversi ?" (ad esempio il primo tasto nero dopo il do è do# (diesis) ma anche re*b* (bemolle)).

... mmmmm!
Non è semplice capire il perché di queste ambiguità, le ragioni della loro origine risalgono ai tempi anteriori alla convenzione sul temperamento (fatto relativamente recente, basta ricordare che J. S. Bach scrisse il *Clavicembalo ben temperato* per appoggiare la causa del temperamento di fronte a chi ne negava la validità).
In quei tempi non esisteva un'unica accordatura delle scale comunemente

Fig. 1.12 La scala diatonica.

accettata, c'erano diverse proposte d'accordatura, e anche diverse accordature delle singole note a seconda che la scala fosse ascendente o discendente.
Tutti convenivano nell'accordatura delle sette note organizzate nella cosiddetta *scala diatonica*, — do, re, mi, fa, sol, la, si, do — ma anche lì i tedeschi usavano invece del *si* un *si* leggermente scordato in basso chiamato *si-morbido*, ovvero secondo la tradizione tedesca di nomenclatura *B-moll* origine anche della denominazione dell'alterazione discendente *bemolle*.
La domanda che immediatamente nasce è: Bene, una volta tutti d'accordo, perché non nominare ogni evento con un nome?
La risposta è anche qui complessa.
Gli eventi sono dodici, però, fino a Schoenberg (all'inizio del nostro secolo) che nel suo sistema dodecafonico rivendicava la possibilità di organizzare in qualsiasi modo i dodici eventi, senza più gerarchia se non l'uso di una base generatrice chiamata *serie* (che permette di usare i dodici eventi in tutte le possibilità combinatorie), venivano organizzati solo in due modi.
Storicamente, sopra i dodici eventi musicali operavano solo due gerarchie di organizzazione: la *modale* e la *tonale*.

1.3.1. *L'organizzazione interna nella musica modale*

Il concetto basilare per capire l'organizzazione modale è il concetto di scala.
La scala è una serie di note, ordinate in progressione sia verso l'alto che verso il basso, con inizio da una nota qualsiasi fino a raggiungere la relativa ottava.
Le scale musicali universalmente usate sono molte, ad esempio:

- la scala pentatonica (di cinque note) usata nelle culture di Cina e Giappone e presente nella cultura europea con un'accordatura diversa;
- la scala *sa-grama* (*grama* in indù significa ottava), molto conosciuta nell'occidente poiché con essa si accordano il sitar e altri strumenti,

presente anche in Europa attraverso la musica zingara e nella parte vocale del flamenco spagnolo;

- la scala *araba a 17 toni*, usata nei paesi del Nord Africa detta anche Egiziana (le diverse culture nel mondo arabo hanno diverse espressioni nella configurazione delle loro scale, infatti solo nel Marocco si usano più di quattro scale diverse, come la "Al-Andalus", la "Al-Magreb", le scale primitive dei Tuareg e la "standard araba" o Egiziana);
- la scala *a toni interi*, in cui le note hanno un rapporto con l'anteriore uguale alla distanza di un tono; è dunque di sei note;
- la scala *fisica*, costruita sulle caratteristiche fisiche del suono, rispettando come generatori i rapporti frazionali esatti tra le frequenze delle note (allo stesso modo di come si producono gli armonici fisici, questa scala ha un suono estremamente monotono).

Le scale usate nel mondo sono moltissime (la maggior parte delle culture umane usa musica di carattere fondamentalmente modale), e so che questo normalmente stimola la curiosità e la creatività dei musicisti.

Nell'informatica musicale è molto semplice lavorare con scale da noi prefissate, ma è anche vero che gli strumenti MIDI e la stessa codifica MIDI sono orientati verso la scala occidentale (cromatica).
Nel secondo capitolo spiegherò come si può usare questa codifica per esprimere e realizzare scale non occidentali.
La scala basilare della musica occidentale è la *scala diatonica*. È costituita da sette note tra loro non equidistanti, ma con una combinazione di rapporti fra toni interi e semitoni (figura 1.13).

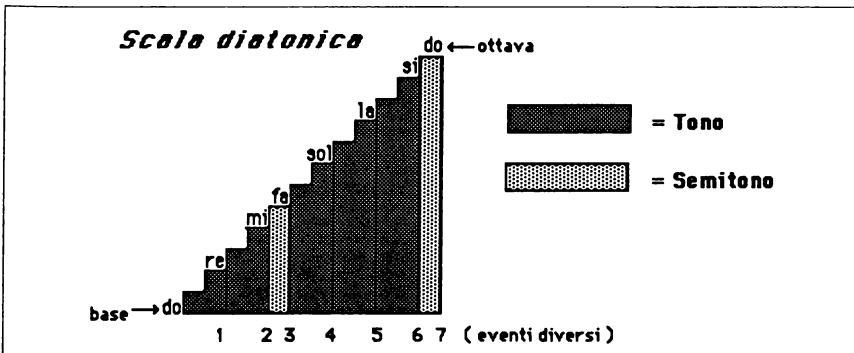



Fig. 1.13

Per estensione il sistema basilare della musica occidentale è un sistema basato sull'operatività come gioco di solo sette note.

1.3.2. *Cerchiamo di costruire un brano modale*

Il sistema modale è un sistema proveniente dalla Grecia antica; è melodico e *non* armonico e si lavora in base ad una scala prefissata. L'unica regola fondamentale è che *il brano deve finire con la nota basilare di detta scala.*
Se vuoi aggiungere un po' di accompagnamento simile all'armonico, si può giocare con l'uso nel basso (o pedale) di una figura ritmica (tipo basso nella musica pop) composta solo dalla nota basilare di detta scala o al massimo un gioco di alternanza di questa con la sua fondamentale (o quinta) finendo comunque con la nota basilare.
È necessario a questo punto che provi a costruire un brano, giocando con i parametri sopraelencati e con una scala già tabulata (o che potrai crearti).

Fig. 1.14 Un esempio di brano modale.

Gia fatto!
Il risultato (tutti ci innamoriamo dei primi risultati sperimentali e queste cose sono una provocazione alla creatività) probabilmente sarà positivo.
Però è senz'altro arcaizzante (non è un peccato) e probabilmente anche un po' monotono (è curioso vedere l'origine semantica di questa parola proveniente della cultura musicale, che significa proprio "lavorare in un solo tono"; per estensione semantica oggi viene intesa come sinonimo di noioso).
Andiamo avanti.
Non necessariamente un brano doveva essere costruito sulla base di una sola scala modale (il tipo di brano fatto solo con una scala si chiama monodico), infatti i greci antichi alternavano normalmente diversi modi nel corso delle loro recitazioni e feste, e molte volte due modi vicini si combinavano in un solo brano.
Però, questo sì, in stretta sequenza ovvero uno dietro l'altro.

Questi "modi" teorizzati da Pitagora e dai suoi allievi (i pitagorici) servivano per diversi scopi, che oscillavano tra la funzionalità (ideale per cantare elegie, per accompagnamento di tragedie, buono per feste dionisiache ecc.) e scopi trascendenti.
I modi rappresentavano, nella loro filosofia, anche un esempio dell'ordine del Cosmo e corrispondevano a diverse assegnazioni mistiche.
Più tardi troveremo sempre la musica, che oscilla dal terreno al mistico.
Nella cultura araba maomettana c'era una corrente che considerava la musica come figlia del peccato; infatti, rispetto alla musica, gli arabi avevano dubbi tra la tolleranza e la diffidenza.
La chiesa cristiana nel secolo XIII disapprovava il "modo ionico" come "modus lascivus" e lo vietava assolutamente nei suoi domini, accettava invece il "modo dorico" e il "lidio" nel suo canto gregoriano.
Comunque la chiesa del secolo XIII non riuscì nel suo scopo purificatore poiché il modo ionico diventò, con piccolissime variazioni di accordatura, la nostra "scala maggiore".

Proviamo un secondo gioco di composizione modale, però questa volta un po' piu complesso.
Tra tutte le combinazioni possibili c'era una combinazione *vincente* che consisteva nell'alternare il modo *basilare* (o di partenza) con il modo *costruito*, a partire dalla scala formata dalla sua terza nota, ovvero dal suo terzo gradino, vedendola non in senso ascendente ma discendente. Nel medioevo il primo si chiamava "modo vero" ed il secondo "modo minore".

Fig. 1.15 Le scale pitagoriche.

Proviamo la melodia composta prima e facciamo il seguente gioco:

a) Prima la scriviamo nel modo originale.

b) Adesso la trasformiamo nel modo minore e questo lo facciamo con la seguente procedura:
trasponiamo tutto quattro note piu in basso; per farlo possiamo aiutarci con una tabella:

do - la
si - sol

la - fa
sol - mi
fa - re
mi - do
re - si
do - la

adesso ascoltiamo tutte e due (una dietro l'altra).

La sensazione più comune di ascolto è che la melodia ripetuta in modo minore perde gioia e acquista in lirismo e malinconia...
Comunque potremmo continuare a giocare per tempo indefinito. Questo è alla base della musica, tutti i musicisti hanno giocato così questi giochi di trasformazione di "oggetti musicali", sono i mattoni con i quali è stato costruito tutto l'edificio della musica.

Abbiamo fatto un po' di musica nuova (o non tanto nuova); sicuramente abbiamo creato delle melodie e lo abbiamo fatto con dei parametri prefissati e il bello è che questi parametri sono diversi da quelli che si usano comunemente, producono quindi effetti diversi dal solito; in più abbiamo operato trasformazioni sul materiale sonoro originario con strumenti formali rigorosi.
Probabilmente il risultato sarà da poco a molto interessante, però attenzione: il risultato può non essere importante, quello che veramente conta di tutto questo è il gioco metodologico.
Adesso continuiamo con il giallo storico, siamo ancora a metà strada, i cinque eventi continuano a essere senza nome...

1.3.3. *Il trionfo del Sole, la scoperta del sistema solare, la gravitazione, l'attrazione dei corpi celesti (Galileo, Copernico, Kepler) ovvero il trionfo del sistema tonale (dalla Mano di Guido al Clavicembalo ben temperato)*

La metafora è senza dubbio una delle meraviglie dell'intelligenza umana, probabilmente arriverà il tempo in cui gli scienziati e gli artisti si chiameranno semplicemente "metaforici".
Può darsi anche che non succeda mai, però è comunque bene mettere in luce che non solo esiste una correlazione filosofica e temporale tra le due parti di questo titolo, ma anche metaforica.

La prima parte fa riferimento al cambiamento di concezione tra Terra-"centro del Cosmo" e Cosmo come insieme di tanti sistemi solari con molteplicità di gerarchie-rapporti e la Terra come parte (anche minuscola) del tutto.
La seconda parte dà solo indicazione del sorpasso del sistema tonale sui sistemi modali.
Il sorpasso del sistema tonale sui sistemi modali non potrebbe essere capito in modo semplice se il lettore non sapesse cos'è la gravità, cos'è un sistema solare (e anche i piccoli rapporti gerarchici tra pianeti e satelliti) e inoltre se il lettore non avesse immaginazione sufficiente per capire l'esistenza di corpi celesti che possono vagare da un sistema a un altro senza appartenere a nessuno dei due.
Per fortuna siamo nel XX secolo e questi problemi appartengono già alla cultura popolare e non più al mondo degli alchimisti, scienziati pazzi e artisti criptici.
Il centro di un sistema tonale è la nota *tonica* (la nota generatrice della scala con la quale è fatto il brano ed anche la nota con cui finisce).
Le scale capaci di generare tonalità sono di due tipi:

- *maggiore* (corrispondente al modo ionico pre-tonale)
- *minore* (con diverse concezioni attraverso la storia tonale)

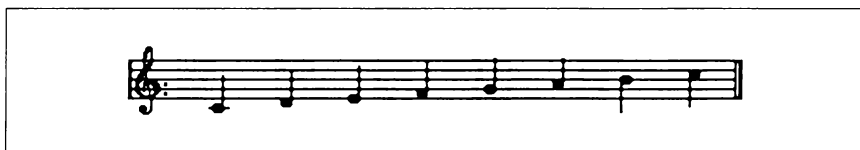



Fig. 1.16 La scala maggiore.

La tonalità è un sistema che non sfrutta solo il rapporto orizzontale tra i suoni (melodico) ma — e questa è la sua vera bellezza — anche un rapporto verticale (armonico).

Fig. 1.17

Fig. 1.18 Gli intervalli.

Quando due suoni vengono messi insieme si produce una tensione sonora assolutamente basilare per la musica, di spiegazione fisica complessa.
La combinazione che si crea da questi due suoni viene denominata dalla distanza che intercorre tra essi.
Le distanze possono essere (per operare con una classificazione preliminare) superiori o inferiori all'ottava. Queste distanze in musica si chiamano *intervalli.*
Quelli inferiori all'ottava in una *scala diatonica* sono:

unisono	do-do (duplicazione della stessa nota)
seconda	do-re
terza	do-mi
quarta	do-fa
quinta	do-sol
sesta	do-la
settima	do-si

La combinazione di tre o più suoni si chiama *accordo.*

Gli accordi sono in realtà tanti quante le combinazione possibili derivanti da tutte le note prese almeno tre a tre (due note non formano un accordo).

Fig. 1.19 L'accordo di do maggiore.

Ritorniamo ancora alle combinazioni di due note. Tra tutte le combinazioni possibili ce ne sono due estremamente "magiche" che sono:

la quinta	do-sol
la terza	do-mi

La quinta definisce l'ambiente tonale; la terza definisce il "modo" della scala generatrice.
La spiegazione per cui la terza definisce la scala generatrice si vedrà provando la stessa combinazione con l'esempio pre-tonale del modo vero (d) e confrontandola con il suo relativo minore (la)

unisono	la-la
seconda	la-si
terza	la-do (terza minore)
quarta	la-re
quinta	la-mi
sesta	la-fa (sesta minore - o napoletana)
settima	la-sol (settima minore o sensibile di dominante)

Fig. 1.20 Intervalli nella scala minore.

È evidente che la terza tra do-mi suona in maniera diversa dalla terza tra la-do: questa differenza è dovuta al fatto che il numero di semitoni tra do-mi è diverso da quello tra la-do (figura 1.21).
La prima terza definisce il modo maggiore (ed è chiamata terza *maggiore*); la seconda terza definisce il modo minore (ed è chiamata *terza minore*).
Invece la quinta do-sol e la quinta la-mi, essendo il loro rapporto uguale (a livello di numero di semitoni), rinforzano il peso della nota bassa e dunque la definizione di tonalità.

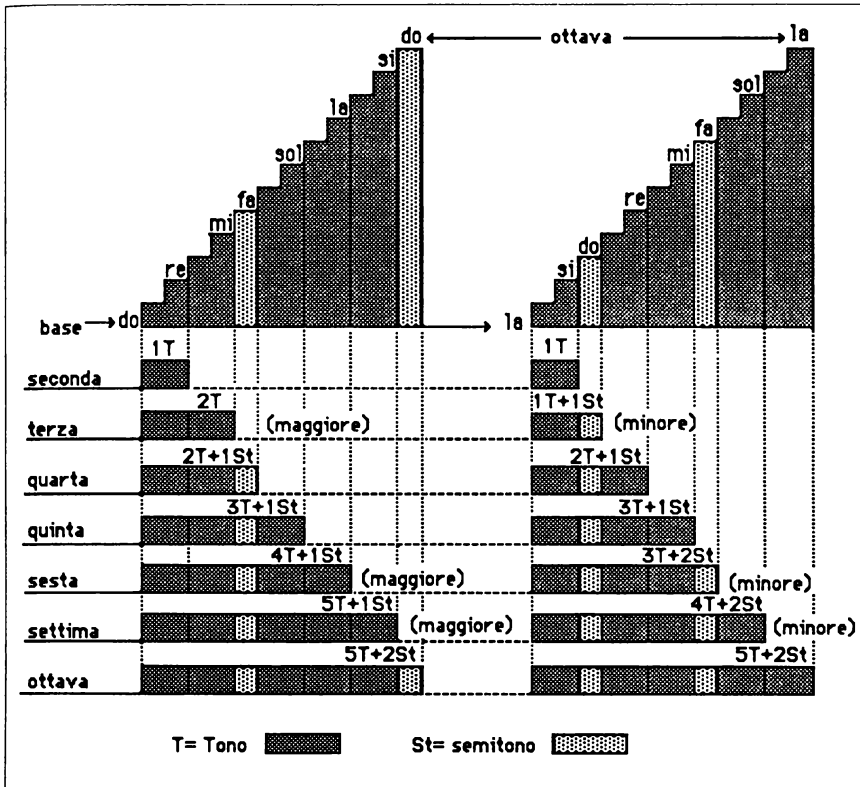



Fig. 1.21 Intervalli in termini di toni e semitoni.

Questo rapporto viene chiamato di *quinta giusta*, invece la nota quinta nella scala diatonica tanto in modo maggiore come minore viene chiamata, per questa sua influenza, la *dominante*.
Esistono altri rapporti che hanno un'importanza decisiva nel gioco armonico: la quarta do-fa è in realtà l'inversione (o rovesciata) di una quinta tra fa-do. In una scala diatonica il rapporto di quarta costituisce, tanto nel modo maggiore quanto nel minore, un'altra invariante (a livello di numero di semitoni) e la nota nella scala viene per questo chiamata la *sottodominante*.

Il resto dei rapporti importanti potete trovarli nei manuali di armonia, soprattutto vi consiglio, se avete voglia e tempo di conoscere questa arte, di con-

tinuare con la lettura di due libri fondamentali; il primo è molto semplice e pieno di esercizi divertenti (inizialmente almeno) ed è il *Manuale di Armonia* di Paul Hindemith, il secondo è più complesso, ma vale la pena di leggerlo, ed è il *Trattato di Armonia* di Arnold Schoenberg.

Adesso, conoscendo i rapporti tra le diverse scale maggiori e minori, cerchiamo di raggiungere lo scopo della ricerca. Ovvero di scoprire ill perché dell'esistenza di questi cinque eventi senza nome, rappresentati nella tastiera musicale dai tasti neri.
Proviamo a generare, a partire dalla dominante o dalla sottodominante di un modo maggiore, un'altra scala diatonica che nasce a partire da quella nota con il processo di trasposizione.
Il processo di trasposizione è un processo molto naturale che avrai utilizzato varie volte non sapendo che si chiamava così.
Supponiamo che tu voglia cantare una canzone o un brano e lo prenda troppo alto; a un determinato momento la voce non ti arriverà più.
Logicamente in quel momento o ti "ammazzi la voce" o ti fermi e riprendi lo stesso brano un po' più basso. Se non sei assolutamente stonato il nuovo brano sarà identico al precedente, però in un altro *tono*.

> *NB1* In questo caso il termine *tono* significa *tonalità* (da non confondere con l'altra accezione del termine, concettualmente molto diversa, usata ampiamente da noi finora per indicare la distanza tra i gradi d'una scala: un tono = due semitoni).

Sei riuscito a fare una trasposizione.
Il motivo per cui il nuovo brano è uguale al precedente è spiegato dal fatto che i rapporti intervallari tra le note che costituiscono i due brani sono gli stessi. Si può dire che l'*algoritmo* (successione di tempi e intervalli) dei due brani è lo stesso, però viene applicato, nella tonalità, partendo da punti diversi.
Logicamente questa è una delle caratteristiche più importanti del sistema tonale: il sistema tonale ha una definizione precisa della accordatura (il la = 440 hertz).
Questo con i sistemi modali non succedeva, il "modo" viene marcato per i rapporti tra i suoi elementi con un'accordatura abbastanza flessibile (in tonalità il do maggiore è totalmente diverso dal re maggiore, mentre a livello modale entrambi sono un "modo Ionico", come qualsiasi scala maggiore).

Dunque, si applica una trasposizione alla scala maggiore partendo, invece che

dal do, dal *sol* o dal *fa* e lo facciamo suonando solo i tasti bianchi (perché fino ad ora sappiamo usare solo quelli)
Partendo dal sol

sol la si do re mi "?" sol

Tutte le note concidono meno il *fa* che, suonandolo con il tasto bianco, suona scordato — in basso — e dunque non è possibile la trasposizione perfetta; proviamo adesso con il tasto nero che c'è tra il *fa* e il *sol*:

sol la si do re mi "fa#" sol ok!

Così nasce il concetto di diesis: il diesis, rappresentato musicalmente con il simbolo #, significa l'alterazione verso l'alto della nota.
Tutte le note con il simbolo diesis suonano un *semitono* più in alto; per estensione, il doppio diesis ## significa che le note ascendono di due semitoni ovvero di un *tono.*

NB2 In questo caso il tono ha l'altro significato (quello della distanza) e non quello della NB1, di "tonalità".

Proviamo adesso con la scala a partire dal *fa* (suonando solo con i tasti bianchi come prima):

fa sol la "?" do re mi fa

Tutte le note coincidono, meno il *si* che suonandolo con il tasto bianco suona scordato — in alto — e dunque non è possibile la trasposizione perfetta. Proviamo adesso con il tasto nero che è tra il *la* e il *si* :

fa sol la "si*b*" do re mi fa

Cosi nasce il concetto di bemolle: il bemolle rappresentato musicamente per il simbolo *b* significa l'alterazione verso in basso della nota .
Tutte le note con il simbolo bemolle suonano un *semitono* più in basso; per estensione, il doppio bemole *bb* significa che le note marcate così discendono da due semitoni ovvero un *tono*. (Vale anche in questo caso quanto detto prima in NB2.)

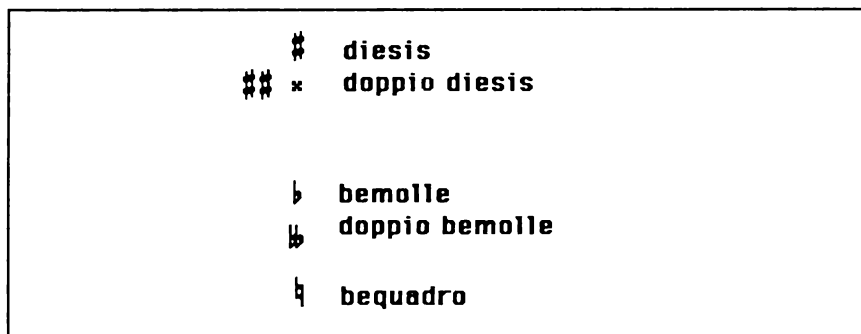



Fig. 1.22 I simboli di alterazione.

È chiaro che si potrebbe nominare il *si b* anche *la#* però è innaturale e scomodo fare una scala con due note con lo stesso nome:

ad esempio *fa sol la la# do re mi fa*

Converrebbe creare un cancellino delle alterazioni, perché per convenzione di scrittura dentro una battuta (e in certi casi che vedremo in seguito all'interno di un intero brano) una volta alterata una nota tutte quelle uguali che la seguono rimangono alterate; se vogliamo restituire la situazione non alterata dobbiamo "cancellare" detta alterazione e per fare questo creiamo il segno di cancellazione o *bequadro.*

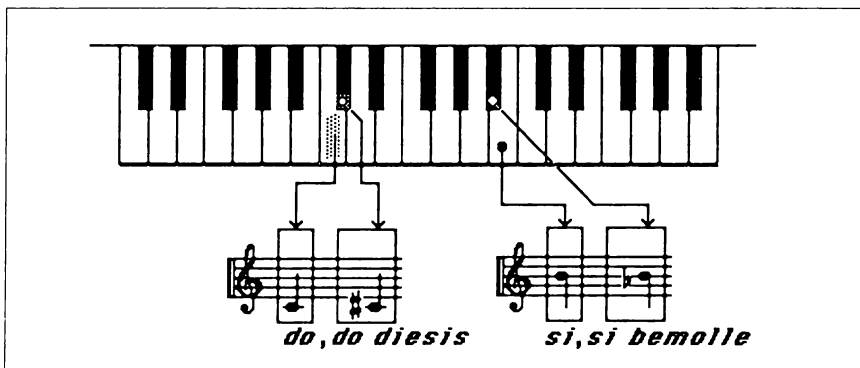



Fig. 1.23

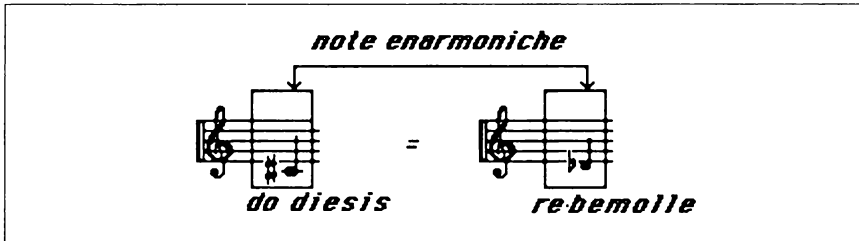



Fig. 1.24

Tutti questi segni che variano l'altezza delle sette note in notazione si chiamano *accidenti.*
Grazie a questi segni possiamo indicare l'uso anche dei tasti neri della tastiera.
Però c'è ancora un piccolo dettaglio da spiegare: come sappiamo già un tasto nero può ricevere una diversa denominazione a seconda che sia diesizzato o bemollizzato e questo sarà così a seconda che appartenga a un tono o a un altro.
Questa duplicità di nomenclatura in musica si chiama *enarmonia* ad esempio il do# ed il re*b* sono due note *enarmoniche.*
Continuando con il processo di generare scale maggiori partendo dalla nota dominante della scala precedente, si procede con la creazione di un famosissimo *circolo* (o *loop*) armonico che si chiama *il circolo delle quinte* (figura 1.25).
La lettura di questo circolo è molto interessante:

- scendendo nello stesso senso delle lancette di un orologio si procede attraverso le quinte, ogni scala è costruita con la quinta della scala precedente;
- scendendo in senso contrario alle lancette di un orologio si procede per quarte, ogni scala è costruita con la quarta della scala precedente;
- i cerchi si ritrovano nel punto di partenza dopo dodici passi; tanti come i tasti diversi nella tastiera musicale;
- la figura centrale piena di tratti rappresenta tutte le possibili combinazione di cambio di tonalità (ovvero *modulazione*) nel cerchio delle quinte;
- procedendo nel senso delle quinte le scale acquisiscono un diesis ogni volta. Per comodità di scrittura questo diesis si scrive insieme alla chiave, ciò significa che *per tutto il brano la nota alterata in chiave verrà intesa come diesizzata;*
  NB.: nel disegno i numeri vicini al rigo indicano il numero esatto di accidenti (1, 2, 3, ecc.)

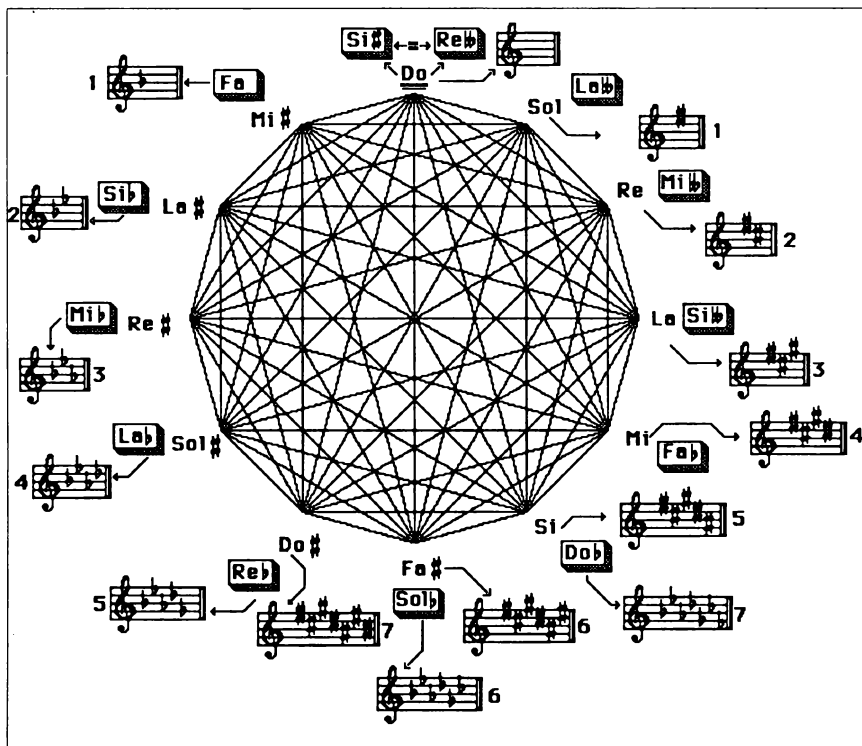



Fig. 1.25 Il circolo delle quinte nei modi maggiori.

- procedendo nel senso delle quarte le scale acquisiscono *un bemolle ogni volta* .
  Per comodità di scrittura (come nel caso precedente) questo bemolle si scrive insieme alla chiave, questo significa che *per tutto il brano la nota alterata in chiave verrà interpretata come bemollizzata;*
  NB.: nel disegno i numeri vicini al rigo indicano il numero esatto di accidenti (1, 2, 3, ecc.)
- dunque: gli *accidenti* vicino alla chiave denotano *la tonalità dil brano*;
- è chiaro che le due note scritte insieme (in ogni angolo dei dodici del circolo) sono uguali, o meglio *enarmonicamente uguali*, per comodità di scrittura; il numero massimo permesso in chiave è di sette accidenti, in teoria — e qualche compositore lo fa — si potrebbe continuare fino a dodici alterazioni in ogni senso.

Fig. 1.26 Armature diverse.

Armatura è il nome che in notazione musicale ricevono gli accidenti scritti vicino alle chiave.

Esiste un altro circolo equivalente a questo che indica il rapporto tra le scale minori; però, prima di mostravelo voglio chiarire la storia delle scale minori nella tonalità, perché se le scale minori se non vengono capite correttamente potrebbero provocare notevoli confusioni a coloro che si avvicinano all'armonia, al contrappunto e ai giochi con la tonalità.

1.3.4. *Le scale minori nella tonalità*

La scala maggiore mantiene la successione dei sette eventi tanto in modo ascendente quato in modo discendente. Invece, nelle scale minori non è affatto detto che gli eventi della scala ascendente e quelli della discendente siano uguali.
Sia in senso ascendente che in senso discendente sono sette, ma non necessariamente gli stessi, qualcuno può cambiare e, a seconda di come cambiano, si formano diverse scale minori.

Il "modo minore" viene generato a partire dalla sesta nota della scala maggiore e da lì nasce il primo equivoco.
Partendo dalla scala diatonica sui tasti bianchi (quella che in maggiore è la scala di do) la scala minore relativa sarebbe:

la si do re mi fa sol la

Questa scala ha come nome: *scala minore naturale.*

Ciò che è giusto a livello di convenzione non lo è nella pratica.
Il settimo grado della scala viene alterato per non fare una scala con la settima

Fig. 1.27 Scala minore naturale.

minore (che a livello armonico dà una sensazione di equivocità tonale; ad esempio la settima minore è la settima che si usa nell'accordo di "settima dominante" ed è la maniera più tradizionale di generare un equivoco armonico-tonale capace di farci passare da una tonalità a un'altra, ovvero modulare in maniera dolce).

La scala minore che si applica realmente è la *scala minore armonica*:

la si do re mi fa sol# la

Per convenzione questo accidente non viene messo nell'armatura ma *nel corso della pagina.*
Dunque il settimo grado di questa scala verrà diesizzato ogni volta dentro la scrittura.
Nel corso del secolo XVIII i compositori trovarono che dal punto di vista "melodico", quando si lavorava in modo minore, il salto tra il sesto e il settimo grado della scala minore armonica era un salto troppo lungo (in realtà è un salto di oltre un tono: un tono e mezzo = 3 semitoni), che per il gusto dell'epoca fu considerato non elegante, dunque conveniva appianare questo salto e il modo più semplice era diesizzare il sesto grado della scala armonica:

la si do re mi *fa# sol#* la

Questa è la scala *minore ascendente melodica.*
Invece, per guadagnare varietà melodica, gli stessi compositori consideravano interessante variare l'andamento ascendente della scala rispetto a quello discendente. Nel modo discendente questa scala sarebbe senza accidenti, ovvero riportata alla scala naturale. Così diviene la scala melodica:

la si do re mi fa# sol# la sol fa mi re do si la

Questa è la scala melodica in modo ascendente e discendente.

Fig. 1.28 La scala minore armonica.

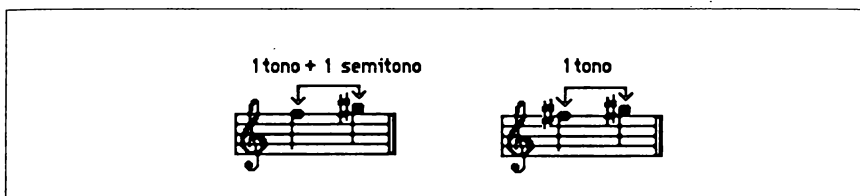



Fig. 1.29 Il sesto grado nella scala minore.

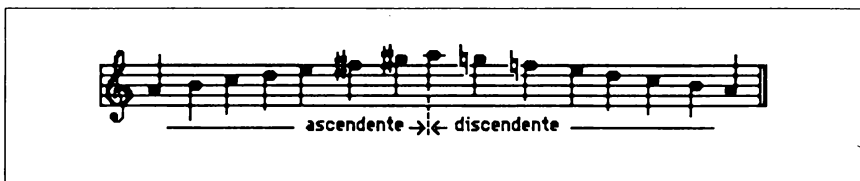



Fig. 1.30 La scala minore melodica.

Qui come succedeva con la scala armonica (per convenzione) gli accidenti non vengono mai scritti nell'armatura ma solo nel *corso della pagina.*
Dunque, per convenzione, i toni minori vengono scritti come se si fossero usate le *scale minori naturali* e questi toni minori tra di loro mantengono dei rapporti simili ai rapporti tra le scale maggiori.

Partendo dal quinto o dal quarto grado di qualsiasi scala minore naturale si procede in circolo di quinte o di quarte (a secondo del senso) fino a chiudere il cerchio.
Nella pagina seguente, in figura 1.31, è riportato il circolo delle quinte per i modi minori.

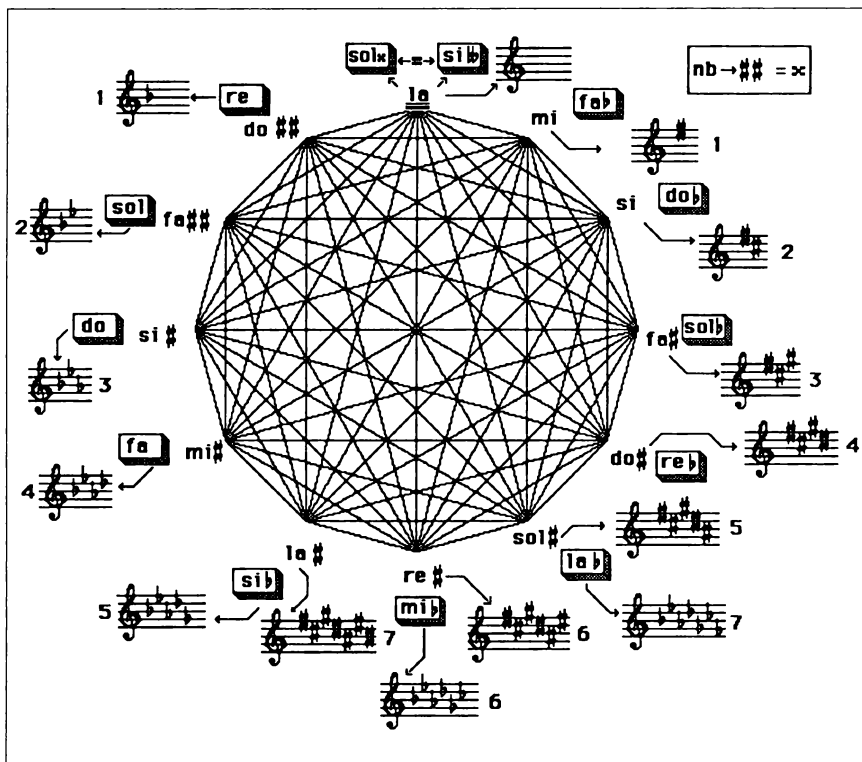



Fig. 1.31 Il circolo delle quinte nei modi minori.

Nella scrittura musicale, quando si parla di tonalità maggiore, questa viene indicata con la prima lettera (o tutte) della nota in maiuscolo, invece le tonalita minori vengono indicate con tutte le lettere minuscole.

La	=	La maggiore
la	=	la minore

Comunque si usa anche esplicitare la minore con una *m* piccola vicina al tono:

la m	=	la minore

Per ricapitolare, l'elenco delle cose importanti spiegate che vi servono per scrivere musica e anche per crearla è il seguente:

per scrivere:

a) il concetto di *notazione*

b) come si scrive l'altezza:
- i dodici eventi (i sette elementi che hanno un nome e i cinque elementi senza nome)
- le pause
- il rigo o pentagramma
- le chiavi e la loro proporzionalità
- accidenti: diesis, bemolle, bequadro
- armatura

c) quali sono i simboli di durata:
- in nomenclatura italiana
- in nomenclatura anglosassone
- la loro equivalenza

per creare musica:

a) il concetto di *scala*
- occidentale e non

b) il concetto di musica modale
- il gioco orizzontale (melodico)

c) il concetto di musica tonale
- il gioco orizzontale e anche verticale (armonia)

1.4. Il suono e la partitura

Ovvero: tutti i concetti che ti mancano per conoscere bene il suono — come materia prima della musica e anche tutti i segni che ti possono permettere di leggere qualsiasi partitura.

Mancano ancora tre elementi molto importanti per capire tutta la potenzialità della notazione musicale:

- il ritmo e la sua notazione
- l'intenzione dinamica e la sua notazione
- gli aspetti interpretativi oltre alla dinamica o al ritmo

Il *ritmo* è un concetto compreso da tutti fondamentalmente in modo intuitivo, si dice che una musica "ha molto ritmo", si parla anche di "ritmo delle onde del mare", o di "ritmo dei battiti del cuore" o di ritmo cardiaco, ecc.
Tentare una definizione di un concetto capito intuitivamente è difficile, normalmente si evita di definirlo e si cerca invece di elencare una serie di attributi precisi per chiarire la sua particolare applicazione.
In musica il ritmo è qualcosa di molto preciso.
È la maniera di giocare con il tempo, ovvero la maniera di organizzare la successione degli eventi (note e pause) in base fondamentalmente alla loro durata e alla loro "intenzione dinamica".
Questo "disprezzo" verso gli altri aspetti (altezza, timbro) dell'organizzazione ritmica, fa di solito ritenere, a torto, che gli strumenti a percussione, di intonazione imprecisa e timbro caratteristico, siano "il ritmo".
Infatti la loro funzione musicale, soprattutto nella musica popolare, è fondamentalmente "ritmica", però il ritmo si trova ovunque nella musica, e non solo in questi strumenti .
Il ritmo può essere ciclico o non ciclico (normalmente lo è).
Può essere capito come parametro assoluto (assoluto può essere ad esempio "tanti battiti per minuto"), come un algoritmo di applicazione al tempo fisico in modo relativo (relativo poiché un ritmo a tre può essere lento o veloce ma pur sempre un ritmo a tre).
Il ritmo musicale è sempre un *algoritmo di applicazione* al tempo fisico in modo relativo.
Il concetto che in musica mantiene il rapporto con il tempo e la sua mediazione reale non è il ritmo ma il *tempo*.
Dunque vediamo prima di tutto il concetto di *tempo musicale*.

Il *tempo* nella notazione musicale definisce la velocita di un brano e le variazioni di velocità durante il brano stesso.
Il modo in qui viene codificata la velocità di un brano è attraverso una delle seguenti parole messe sopra il rigo all'inizio del brano.

I tempi generali d'interpretazione in musica
(da meno a più):

Grave
Lento
Largo
Larghetto
Adagio
Andante
Moderato
Allegretto
Allegro
Vivace
Presto
Prestissimo

Per "raffinare" questi indicatori generali del tempo esistono queste indicazioni (ne esistono altre meno usate):

giusto
assai
molto
con moto
sostenuto
ma non troppo
con fuoco

Logicamente si aggiungono al tempo e raffinano la sua definizione. Non sono lo stesso, ad esempio, un *Andante "giusto"* e un *Andante "con moto"* ecc.
Comunque dopo l'apparizione del "metronomo" nell'insegnamento musicale, ci sono molti compositori che preferiscono la definizione del tempo attraverso il rapporto preciso con il metronomo.
L'indicazione si fa dando un numero come valore di semiminima, o minima, o qualsiasi altra figura di durata.

Il numero indica la quantità di figure di quella durata che suonano al minuto, ad esempio:

semiminima = 96 **tempo metronomico** ♩ = 96

Questo significa che ci saranno 96 semiminime al minuto; se il brano è fatto in battute di 4 semiminime sarano 96/4 battute per minuto ovvero 24 battute al minuto.
Le variazioni del tempo vengono segnate in lingua italiana sopra (o sotto) il rigo nel momento in cui si vuole operare la variazione. Le indicazioni più usate sono:

più allegro
meno mosso
accelerando
stringendo
rallentando
ritenuto
rubato

Quando si vuole restituire la situazione iniziale compare l'indicazione "a tempo" oppure "tempo primo", o "tempo I".
L'unico segno che abbia influenza sul tempo nella notazione musicale è la *corona.*
Una "nota con corona", dura una quantita di tempo arbitraria in più, e tutto il brano si ferma ad aspettare che finisca.
La corona (curiosa affermazione del potere monarchico dell'epoca) significa infatti una "congelazione" del tempo al servizio di quell'evento.
Adesso vediamo i giochi ritmici più importanti in musica e la loro codifica nella notazione musicale.
Prima abbiamo visto come scrivere le durate degli eventi musicali (note e

Fig. 1.32 Nota con corona.

pause), adesso due segni importanti che ci mancano:

- i *puntini*:
  dopo la figura un puntino indica che il valore (durata) della figura viene allungato di metà del suo valore, ad esempio:

  =

  =

  un doppio puntino indica che il valore (durata) della figura viene allungato di una volta e mezza del suo valore ad esempio:

  =

  =

- *la legatura di valore* o *"tie"* (termine anglosassone):
  due note della stessa altezza unite per una legatura (arco sopra o sotto) in realtà sono un unico evento di durata equivalente alla somma delle due figure. (Vedi l'illustrazione sopra.)
  Il motivo di questa notazione è doppio:
  - nel caso in qui l'evento si produce tra due battute è l'unico modo di scriverlo rispettando la convenzione della battuta (se ricordate, l'unità basilare della notazione è la battuta)
  - se l'evento appartiene a una sola battuta e la sua durata non è (1 + 1/2) i puntini non bastano ed è neccessario costruirlo attraverso la legatura di due figure.

Avete visto questi due segni che vi mancavano per capire tutto quello che è neccessario per scrivere correttamente in "notazione" qualsiasi o un evento con qualsiasi durata (superiore o inferiore ad una battuta).

1.4.1. *La scrittura ritmica*

Il ritmo è fondamentalmente *l'organizzazione degli accenti.*
Usando la metafora del linguaggio naturale, tutte le parole con più di una sillaba hanno un accento, questo succede anche in musica.
L'*accento* in realtà non è che un aumento di "intenzione dinamica" nella realizzazione di un determinato evento sonoro (logicamente le pause non si possono accentuare) che come risultato genera un volume superiore in quell'evento.
L'applicazione sistematica degli accenti genera durante l'esecuzione di un brano un gioco di tensione-rilassamento che dà un senso ritmico al discorso musicale.

Per convenzione la prima nota di ogni battuta ha un accento.

Il gioco degli accenti è una delle fonti d'ispirazione musicale più ricca nella storia, non solo nella cultura europea ma in ogni cultura musicale.

La gente normalmente fa confusione nella lettura della "misura" e fa confusione fondamentalmente perché la notazione è numerica, trappolosa, perché non è evocativa. Voglio approfondire questo argomento, per chiarire concetti musicali basilari che non sempre sono ben capiti.
L'accento, in notazione, raramente viene evidenziato in modo esplicito.
Nella notazione musicale l'accentuazione è *implicita* a seconda dell'organizzazione ritmica del brano e, dunque della sua misura.
In notazione si riassumono tutte le possibilità di accentazione alla combinazione di due sole primitive.

Gli accenti "standard" sono ogni *due* o *tre battiti.*

Il primo sistema di accentazione costituisce i gruppi chiamati *gruppi binari.* I

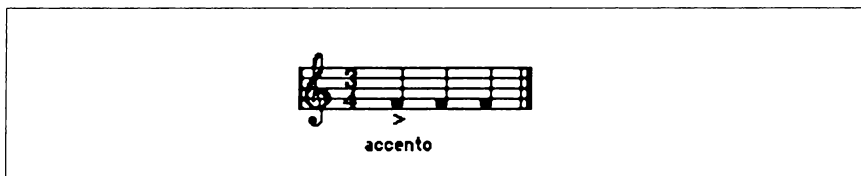



Fig. 1.33 Accento.

Fig. 1.34 Accenti e gruppi.

battiti sono raggruppati a due a due, con un battito forte alternato a un battito debole.
Il secondo sistema di accentazione genera i gruppi chiamati *gruppi ternari.* I battiti sono raggruppati tre a tre, con un battito forte alternato a due battiti deboli.
La notazione musicale *non evidenzia* su quale primitiva o primitive ritmiche viene scritto il brano; la conoscenza esatta di queste primitive può permettere all'utente (musicista) di ricostruire l'organizzazione "vera" degli accenti nel brano a partire delle indicazioni generali di misura o "meter"(termine anglosassone).
La *misura* non è un'organizzazione di "capacità delle battute" — due, tre, quattro, cinque, ecc. semiminime a battuta — ma è *un'indicazione generale di come costruire in maniera "standard" gli accenti ritmici del brano.*
La misura viene indicata all'inizio del brano nel rigo mediante una frazione; il cambio di misura viene evidenziato con l'apparizione di un'altra frazione nel

Fig. 1.35 Indicazioni di misura.

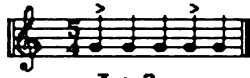
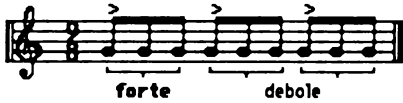
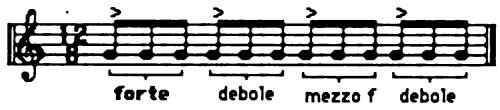



Fig. 1.36 Misure e accenti.

momento in cui si vuole cambiare di misura (per convenzione *sempre all'inizio di una battuta*, dunque non è possibile cambiare di misura alla metà di una battuta).
La frazione indica due cose:

- il numeratore indica il "tipo di misura" (dunque, dal suo valore si sopra capire poi l'organizzazione standard degli accenti e il numero di battiti per battuta)
- il denominatore indica il tipo di durata del "battito".

Dunque la frazione non è "una frazione e... basta", come tutti potrebbero immaginare (questo penso sia all'origine di tutte le confusioni) ma è l'indicazione di un rapporto complesso tra accenti e durate dei battiti. (La figura 1.36 può essere d'aiuto.)

Se il numeratore è 2

La misura è formata da un solo gruppo binario dunque l'accentazione è: primo battito forte secondo debole, stanghetta, primo battito forte secondo debole, ecc.
Ovvero: due battiti per battuta, con l'accento sul primo dei due battiti.

Se il numeratore è 3

La misura è formata da un solo gruppo ternario dunque l'accentazione è: primo battito forte secondo e terzo deboli, stanghetta, ecc.
Ovvero: tre battiti per battuta, con l'accento sul primo dei tre battiti.

Se il numeratore è 4

La misura è formata da due gruppi binari, il primo più forte del secondo, dunque l'accentazione è: primo battito forte secondo debole terzo forte (però meno forte del primo) quarto debole, stanghetta, ecc.
Ovvero: quattro battiti per battuta, con due accenti, sul primo e sul terzo battito di ogni battuta.

Se il numeratore è 5

La misura è formata da due gruppi uno binario l'altro ternario, l'ordine non viene predeterminato — può essere 2 -3 ma anche 3 - 2 — dunque l'accentazione è: un accento sul primo battito del gruppo binario e un altro sul primo battito del gruppo ternario a seconda della loro organizzazione (2-3 o 3-2) l'accento più forte dei due sarà sempre quello che

coincide con il primo battito della battuta, ogni cinque battiti stanghetta, ecc.
Ovvero: cinque battiti per battuta, con due accenti sui primi battiti dei due gruppi (uno binario l'altro ternario) che compongono la misura.
Questa misura è chiamata *irregolare* (insieme alla misura a 7); con la sua ambiguità di rappresentazione (2-3 o 3-2) denota la mancanza di univocità della notazione musicale.

Fino qui la teoria dei gruppi binari e ternari come generatori della misura e degli accenti si svolge in modo lineare.
Esistono altri tipi di misure chiamate *composte* perché si ottengono per combinazione delle misure elementari sopra esposte non in maniera lineare ma a livelli diversi.
Vediamo come si costruiscono le misure composte facendo un esempio: se ad una "misura a due tempi" (forte - debole) sostituiamo un solo evento (battito) per un gruppo binario o ternario potremmo ottenere una combinazione d'intenzione così:

battito 1	battito 2
forte	debole
gruppo 1	gruppo 2

Permangono sempre gli accenti caratteristici del gruppo con la condizione ulteriore che il primo gruppo è più forte del secondo.
L'effetto (usando una metafora) si potrebbe paragonare all'intenzione della "misura a due tempi", che opera come *inviluppo* della sequenza dei gruppi (binari, ternari o irregolari).
Detto questo possiamo continuare con il commento alla figura 1.36.

Se il numeratore è 6

La misura è *composta* e proviene da una "misura a due tempi", è però formata da due gruppi ternari, il primo più forte del secondo; o tre gruppi binari, il primo dei quali più forte degli altri due.
Dunque nel primo caso l'accentazione è: i tre primi battiti forti, quarto, quinto e sesto deboli, però il primo e il quarto leggeramente accentati rispetto agli altri battiti del loro gruppo, stanghetta, ecc.
Nel secondo caso l'accentazione è: i due primi battiti forti e gli altri quattro deboli, però il primo il terzo e il quinto leggermente accentati rispetto agli altri battiti del loro gruppo, stanghetta, ecc .

Ovvero: la battuta divisa in una prima metà forte ed una seconda metà debole con sei battiti per battuta e con due accenti, nel primo e nel quarto battito di ogni battuta.

NB. Le misure a 6/8 *non devono mai essere intese come tre gruppi binari* (questo logicamente per convenzione)

Se il numeratore è 7

È una misura "irregolare" ed è formata da tre gruppi, due binari l'altro ternario (l'ordine non viene predeterminato, può essere qualsiasi loro combinazione: 4-3, 3-4 oppure 2-3-2) l'accentazione naturalmente dipende della loro configurazione nel brano.

Ovvero: sette battiti per battuta, con tre accenti essendo il più forte dei tre quello che coincide con il primo battito della battuta.

Se il numeratore è 8

Non si usa (perché è una ridondanza della misura a 4).

Se il numeratore è 9

Questa è un'altra "misura composta"; proviene da una "misura a tre tempi" formata da tre gruppi ternari, il primo più forte degli altri due.

Se il numeratore è 10

Non si usa (perché è una ridondanza della misura a 5). Comunque l'ambiguità delle misure irregolari qui verrebbe ampliata.

Se il numeratore è 11

Non si usa; nel caso in cui si volesse usare, si può procedere analogamente come per le altre misure irregolari.

Se il numeratore è 12

La misura a dodici può essere intesa come una "misura composta" a due tempi o a quattro tempi.

Ancora di più qui che nelle misure a 6 o a 9 impera il senso di *envelope* della misura generatrice (e non dei suoi gruppi binari o ternari); infatti se la si considera come generata da una misura a due tempi, sarà vista come con sei battiti forti e sei battiti deboli, se la si considera invece generata da una misura "a quattro tempi" (ed è il modo comune di intenderla), il primo gruppo ternario sarà forte, il secondo debole, il terzo forte — però

meno forte del primo — e il quarto debole (allo stesso modo dei battiti nella misura a quattro tempi).

La tabella può continuare “ad libitum”, tuttavia quelli sopra esposti sono i valori più usati; soprattutto abbiamo passato in rassegna tutti i concetti basilari per l’eventuale costruzione delle altre “misure”.

Per ricapitolare, quei concetti basilari sono:

- *Primitive di costruzione (gruppi a due e a tre):*
  - Misura a due
  - Misura a tre
- *Combinazione delle stesse primitive in modo sequenziale:*
  - Misura a quattro
  - Misure irregolari
- *Combinazione delle stesse primitive a diversi livelli:*
  - Misure composte

È libera la possibilità di creare ulteriori tipi di misure, e il tentativo di crearli può essere piacevole e interessante, però un’altra organizzazione degli accenti dovrà essere accuratamente specificata.

Il *denominatore* nella frazione indica il tipo di durata del “battito”:

1 - semibreve
2 - minima
4 - semiminima
8 - croma
16 - semicroma
32 - biscroma (non si usa)

La lettura diretta del rapporto rappresentato dalla frazione indica il numero di battiti per battutta. Per esempio:

3/1 = 3 battiti di semibreve per battuta
2/4 = 2 battiti di semiminima per battuta
5/8 = 5 battiti di croma per battuta

A questo punto si hanno tutte le informazioni per costruire qualunque

combinazione ritmica in qualunque brano.
Nel campo del software musicale, tutti i programmi che implementano il concetto della misura, scrivono accenti, e almeno accentano la prima nota di ogni battuta; altri più evoluti implementano tutti gli accenti e lo fanno in modo standard (soprattutto i programmi di traduzione dalla scrittura notazionale alla MIDI).
È necessario capire questi concetti perché i disegnatori dei software musicale li applicano sistematicamente.
Ad esempio può succedere che un programma trasferisca un brano scritto con la notazione alla MIDI, il brano si ascolta e vediamo che gli accenti sono in modo sbagliato rispetto a come noi volevamo, questo sicuramente sarà perchè probabilmente è stata usata una misura sbagliata o il programma quando riconverte in Midi non acetta cambi di misura nel corso del brano (problema gravissimo causato da una deficiente concezione delle informazioni MIDI).

Per finire, in musica la regolarità è bella, ma anche l'irregolarità e bellissima.
Il ritmo non è un tutto deterministico e ossessivo (ma lo può anche essere).
Dunque, esistono dei giochi ritmici basati sulla disparità della durata degli eventi.
All'interno di battiti regolari accade spesso che vi siano raggruppamenti irregolari di note (o eventi). Sono i cosiddetti *gruppi irregolari* (*tuplets* in inglese).
I più usati sono:

2 invece di tre ovvero *duina*
3 invece di due ovvero *terzina*
4 invece di tre o sei, ovvero *quartina*
5 invece di quattro o di tre ovvero *quintina*
6 invece di quattro ovvero *sestina*
7 invece di quattro o di sei ovvero *eptina*
8 invece di nove
9 invece di otto
10 invece di nove o di otto
11 invece di otto o di nove
13 invece di dodici, otto o nove

Si scrivono unendo tutte le note che compongono il gruppo con un archetto su cui è segnata la cifra rappresentativa del loro valore (figura 1.37).
Nel software musicale occorre definire esattamente la frazione (molti software

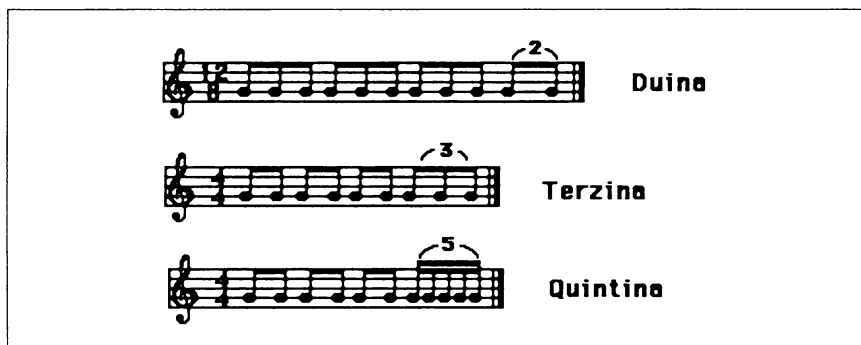



Fig. 1.37 Gruppi irregolari (*tuplets*).

accettano un numero molto limitato di gruppi irregolari): ciò è dovuto a ragioni di implementazione software.
Arrivati a questo punto, il mio consiglio è di usare l'Atari ST, con un qualunque programma di scrittura o un sequencer, immettendo dati step by step, per provare ad "ascoltare " i tuoi algoritmi ritmici sfruttando le idee sopraesposte. È importante imparare a scrivere le proprie idee, però secondo me è più importante poterti fornire concetti chiari e idee nuove per stimolare la tua creatività .
A livello ritmico esiste un concetto basilare, che in notazione musicale non compare, perché non è stato codificato: la notazione tradizionale ha come limitazione quella di lavorare a un unico livello di astrazione, non permettendo tante volte di lavorare con gli "oggetti musicali", autentica fonte della creazione musicale.
Il concetto non esistente in notazione è quello di *pattern* (ovvero "modello"). Il *pattern ritmico* è un oggetto musicale che rappresenta ad un qualsiasi livello di astrazione — anche oltre la battuta — l'organizzazione di un *gruppo qualsiasi* di eventi ritmici.
Dato che è un concetto che opera a più livelli di astrazione prescindendo dalla nozione di battuta, la notazione lo sopporta con difficoltà; si può comunque cercare una rappresentazione notazionale adeguata.
Il pattern è una struttura che si ripropone nel corso del brano una o più volte. Ci sono pattern a diversi livelli di astrazione: per dirla in altro modo, un intero pattern può essere considerato come una delle primitive di un pattern di livello superiore, o anche: ogni pattern è l'organizzazione di pattern di livello inferiore.

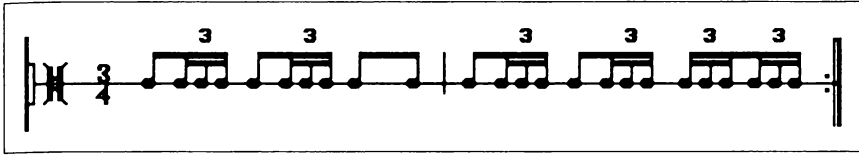
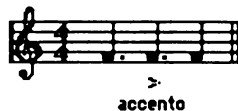



Fig. 1.38 Pattern ritmico del *Bolero* di M. Ravel (1928).

Ad esempio:

i gruppi binari o ternari sono un pattern
le misure semplici sono un altro pattern

Le misure composte sono un pattern di livello superiore composto da pattern di livello inferiore (i gruppi) strutturati seguendo il "modello" o "pattern" delle misure semplici.
In notazione vedremo che ci sono "pattern ritmici" composti da oggetti musicali di durata superiore alla battuta
Il pattern ritmico del *Bolero* di Ravel è senza dubbio uno dei pattern più conosciuti della musica di questo seccolo. La sua durata è di due battute e si ripropone in "loop" (ciclicamente) per tutto il brano.
Ci sono anche pattern di durata inferiore alla battuta. La sincope, per esempio:

è senza dubbio un pattern ritmico d'importanza fondamentale nella storia della musica, e si trova nella musica di tutte le culture umane. Il suo effetto è di rottura della accentazione standard, con il trasferimento dell'accento sui tempi deboli.
L'uso di macchine tipo sequenziatori e drum machine ha familiarizzato i musicisti con l'uso dei pattern. Ciò non è dovuto a una concezione "avanguardista" dei fabbricanti, ma alla poca memoria disponibile in questo tipo di macchine: per fare un paragone, un sequencer di drum macchine può usare al massimo 64 K per tutto (gestione e memoria), contro i 512 o i 1040 K di un Atari ST.
I chip di memoria fino poco tempo fa erano molto costosi e i fabbricanti cercavano con tutti i mezzi di risparmiare memoria per diminuire i costi di

fabbricazione. Oggi questi chip si stanno abbassando di prezzo sensibilmente e questo, senza dubbio, favorirà la comparsa sul mercato di prodotti sempre più potenti per quel che riguarda la loro capacità di memoria.
L'uso dei pattern comporta un risparmio importante di memoria nella macchina, in quanto non occorre più memorizzare tutte le note ma solo le varie occorrenze dei pattern o al più quelle sequenze di note non riconducibili a pattern (ovvero quelle che non si ripetono). In altre parole: una macchina che usa l'organizzazione per pattern deve memorizzare solo poche sequenze per esteso — i pattern, per l'appunto.
Dopo serve solo un'unica sequenza di organizzazione dei pattern tra di loro, il pattern dei pattern, ovvero la *song*. Logicamente la ripetizione non occupa in memoria più che un numero, il numero indicativo delle volte che si deve ripetere. Dunque: il risparmio di memoria è enorme.
L'organizzazione attraverso pattern è pratica e semplice e offre all'utente la possibilità di "razionalizzare" il suo lavoro; per questo diventa estremamente comoda ed educativa. Però l'implementazione restrittiva dei pattern, che si trova sulla gran parte dei programmi commerciali, può non fare percepire la validità della "metafora" a tutti i livelli di astrazione, perché si confonde la potenzialità evocativa del concetto di "modello" con l'uso di un semplice "comando" a un solo livello.
Infatti la bellezza concettuale del comando "pattern" viene abbinata in questi prodotti commerciali con la banalità concettuale del comando "song", quando in realtà la song non è che un altro pattern a un livello superiore.

1.4.2. *Dinamica*

Per poter capire il ritmo abbiamo accennato al concetto di "intenzione dinàmica". Adesso approfondiamo come si può scrivere la intenzione dinàmica nella notazione musicale.
Fermiamoci un po' sulla natura fisica degli eventi sonori.

Il suono è un fenomeno fisico ondulatorio prodotto in un intervallo tra 20 e 20.000 hertz. (Di tutte le onde che ci circondano, l'uomo è capace di percepire solo quelle prodotte tra queste due frequenze: il concetto di suono è un concetto umano, culturale; "tutto quello che è udibile è suono ", e tutti i fenomeni fisici uguali a quello del suono che si producono oltre o sotto le possibilità di percezione degli uomini sono chiamati "ultrasuoni" o "infrasuoni".)
L'onda sonora riprodotta nella figura 1.39, che corrisponde al suono di un

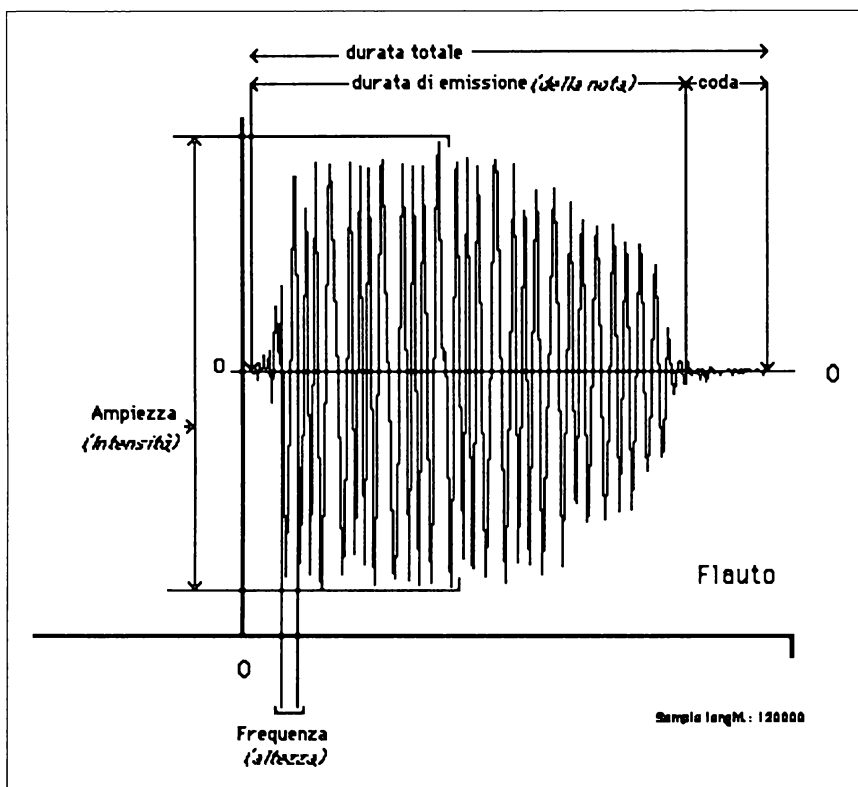



Fig. 1.39 Una forma d'onda.

flauto, come qualsiasi altra onda ha quattro caratteristiche fisiche che la definiscono completamente:

Altezza	numero di volte che si ripropone l'onda completa al secondo ovvero la sua "frequenza", misurabile in hertz (cicli al secondo)
Durata	la durata dell'evento nella sua totalità, misurabile in unità di tempo
Intensità	corrisponde all'ampiezza della vibrazione misurabile in decibel
Timbro	che è proprio la sua "forma d'onda" rappresentabile anche mediante il suo spettro armonico (secondo la teoria di Fourier).

Sappiamo già come scrivere l'altezza e la durata degli eventi musicali;

vediamo ora le convenzioni di scrittura della *dinamica* che è il modo di esprimere musicalmente il giocco con il fenomeno fisico dell' *intensità sonora*:

Non esiste nella notazione musicale per l'intensità un parametro assoluto come lo è il tempo per le durate e il ritmo, o come lo è la convenzione di temperamento per l'altezza (la = 440 hertz).
In musica si codìfica "l'intenzione dinamica" in modo relativo e molto ambiguo. Ad esempio a uno stesso livello d'intenzione dinamica ci sono strumenti di sonorità molto più "potente" che altri; solo nell'orchestrazione si possono trovare delle regole (ed anche lì molto elastiche) di rapporto tra le "intenzioni dinamiche" dei singoli strumenti. Questa ambiguità notazionale per la musica scritta in notazione e dopo realizzata con strumenti elettronici digitali genera delle difficoltà — di traduzione — come vedremo nel capitolo sulla MIDI (concretamente per quanto riguarda la "key velocity").

La codifica della dinamica in notazione

La dinamica nella codifica musicale viene espressa con una serie di segni chiamati *segni dinamici.*
Questi segni vengono scritti di solito sotto il rigo, indicando con quale intensità le note devono essere suonate. Normalmente in notazione tradizionale si accettano come minimo tre ***p***, e come massimo tre ***f***, ma ci sono autori, soprattutto verso la fine del secolo scorso (gli impressionisti) e nella musica contemporanea, che scrivono fino a cinque ***p*** e cinque ***f***.
I segni dinamici sono in crescendo:

ppp	=	più che pianissimo
pp	=	pianissimo
p	=	piano
mp	=	mezzo piano
mf	=	mezzo forte
f	=	forte
ff	=	fortissimo
fff	=	più che fortissimo

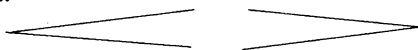
In musica il passaggio da una situazione d'intenzione dinamica a un'altra può avvenire in due modi: improvvisamente o gradualmente.
Il *cambiamento improvviso* si può fare in diversi modi (usando anche delle

indicazioni speciali):

a) Scrivendo il segno dinamico d'intenzione sulla prima nota del cambiamento: cosi si cambia improvvisamente tra due livelli qualsiasi.

b) Con le indicazioni "più" e "meno"; ad esempio: su un passaggio in *f* l'indicazione "più" significherebbe "più forte" — queste indicazioni sono una sfumatura, un "più" o un "meno", non un salto nella gradazione dinamica ma operano, per così dire, "a parità di livello".

c) Con la combinazione di due segni ed anche con l'uso dell' indicazione "subito" ("sub."). Le combinazioni più comuni sono *fp* (forte piano), che vuol dire forte seguito improvvisamente da piano e *pf* (piano forte) che vuol dire piano forte ed è l'effetto inverso del precedente. (Il forte piano è un effetto bellisimo, molto difficile da realizzare bene, perché in molti casi si usa per generare una "risposta allontanata" al forte, ovvero una specie di "eco", in altri casi per "ammazzare la coda" al suono forte. Il suo uso corretto è comprensibile solo analizzando il contesto del brano.) L'indicazione di "subito" marca decisamente il carattere "improvviso" del cambio dinamico.

Il *passaggio graduale* da un livello a un altro viene di solito indicato da un segno a forma di forcella:

Lo si può esprimere anche a parole:

a) Crescendo (*cresc.*)
b) Decrescendo, Diminuendo (*decresc.*, *dimin.* o *dim.*) e anche Morendo o Smorzando.

Altri segni d'interpretazione dinamica

Vi sono una serie di segni usati per indicare effetti particolari di "intenzione dinamica". I più frequenti sono quelli che seguono.

sforzando o *sforzato* (*sf* o *sfz*): rinforzare il suono con accento marcato
staccato : suono breve però marcato
staccatissimo: uno "staccato" ancora più breve
tenuto: la nota deve essere tenuta o sostenuta per tutta l'esatta durata del suo valore e, in più, leggermente accentuata.

L'indicazione più importante di tutte a livello espressivo è il *legato di espressione* (*slur* nella terminologia anglosassone): quando diverse note di diversa altezza vengono unite con un archetto o "legatura", questo significa che vanno eseguite come un "tutto unico", con levigatezza.

1.5. Il timbro e la sua notazione

Il timbro è la quarta caratteristica del suono. Si ricorre sempre ad un esempio standard per spiegare meglio che cosa è: due do3 (C3) suonati uno con un pianoforte e l'altro con un flauto, sono evidentemente diversi anche se sono alla stessa altezza, hanno la stessa durata e anche la stessa intensità; ciò che li rende diversi è "che hanno un timbro diverso".
Il timbro è proprio la caratteristica più "personale" degli eventi sonori perché è la loro "forma d'onda".
Secondo la teoria di Fourier, ogni timbro può essere visto come la somma (in proporzioni variabili) di onde sinusoidali le cui frequenze sono un multiplo di una frequenza fondamentale.
Queste onde sinusoidali primitive di qualsiasi suono sono chiamate *armonici*.
Gli armonici sono di due tipi: superiori e inferiori. Gli *armonici superiori* si ottengono moltiplicando la frequenza base (f) per numeri interi; per esempio:

nota fondamentale = $f \times 1$
1° armonico = $f \times 2$
2° armonico = $f \times 3$
ecc.

Esempio: gli armonici superiori del la con frequenza f = 440 hertz saranno:

fondamentale = $f \times 1 =$ 440 hertz
1° armonico = $f \times 2 =$ 880 hertz
2° armonico = $f \times 3 =$ 1320 hertz
ecc.

Gli *armonici inferiori* ovvero *sottoarmonici* li otterremo dividendo la frequenza di base per numeri interi, ad esempio:

nota fondamentale = f / 1
1° sottoarmonico = f / 2
2° sottoarmonico = f / 3
ecc.

considerando sempre l'esempio del la = 440 hertz si avrà:

nota fondamentale = f / 1 = 440 hertz
1° sottoarmonico = f / 2 = 220 hertz
2° sottoarmonico = f / 3 = 146, 666... hertz
ecc.

Dunque la differenza timbrica tra due suoni (accettando la teoria di Fourier) potrà esprimersi attraverso la differenza della loro composizione "armonica". In notazione *non esiste* una codifica generale del timbro e la sua manipolazione, le variazioni generali di timbro in musica avvengono attraverso l'assegnazione dell'interpretazione di quella "parte" a uno strumento tipo.

Il "documento" musicale basilare si chiama *partitura*: il termine "partitura" sta generalmente a indicare la presentazione scritta della musica eseguibile da un "complesso" (vocale, da camera, sinfonico, jazz ecc.) disposta in modo tale che il lettore sia in grado di vedere "contemporaneamente" tutte le "parti". Da una partitura normalmente si estraggono le "parti" (dei singoli strumenti): in queste parti compare una serie d'indicazioni notazionali direttamente connesse con la "tecnica di suonare" quello strumento e alcune di esse fanno direttamente riferimento alle caratteristiche timbriche dello strumento. Le indicazioni timbriche sono la parte della notazione attualmente meno evoluta e dunque più insufficiente, però *non* esiste alcuna altra codifica per esprimere il fenomeno.

Per finire questa sezione dedicata in modo abbastanza esauriente alla notazione musicale si riporta in figura 1.40 un elenco non solo dei segni particolari di interpretazione (per famiglie di strumenti) ma anche delle altre figure notazionali di carattere generale che non sono state finora esposte perché non hanno carattere fondamentale né sono direttamente vincolate agli argomenti sovraesposti.

Fig. 1.40 Altri segni notazionali (da: Pozzoli, *Sunto di teoria musicale*, Ricordi).

1.6. Cos'é il CAI - prospettive educazionali per la famiglia degli Atari ST

CAI è una sigla che corrisponde alle iniziali in inglese di "Computer Aided Instruction" che tradotto in italiano significa: Insegnamento aiutato dall'elaboratore.
Che l'uso dell'elaboratore ti obblighi a imparare molte cose è ovvio. Però non è questa capacità educativa "in sé" del lavoro con l'elaboratore, ciò che viene inteso come CAI.
CAI è l'etichetta di una linea di prodotti fondamentalmente software — abbinati a periferiche specifiche hardware — orientati all'insegnamento.
La scuola si sta "informatizzando" ed è giusto che sia così, l'informatizzazione "democratizza" e decentralizza la qualità dell'insegnamento.
È chiaro che il rapporto professore-allievo è salvo in questa nuova maniera d'insegnare, perché il professore si libera delle funzioni ripetitive (noiose) e dunque guadagna spazio nel rapporto umano con l'allievo (insostituibile da una macchina).
Comunque nel campo musicale i programmi d'insegnamento con tecniche CAI esistono da poco tempo.
I pioneri in questo settore sono stati i programmi degli israeliani dell'università di Gerusalemme, i giapponesi dell'università di Osaka, la fondazione "Yamaha" e le università americane (Stanford, California; MIT, Massachusetts; CMU, Berkley Music, Boston; ecc.).
Oggi, oltre a questi, esistono già numerosi programmi CAI sviluppati da pubbliche istituzioni e privati, distribuiti da software-house europee nordamericane e canadesi.
Logicamente tutti i programmi sviluppati fino ad ora in questo settore sono stati implementati su elaboratori particolari.
La Yamaha sui suoi CX5, gli israeliani su macchine MS-DOS (PC e compatibili), Apple II, Macinstosh, e Atari ST, sfruttando delle interfacce speciali.
Non esiste finora nessun programma CAI d'insegnamento della musica accettato come sistema "standard" in nessun paese del mondo.
Tutti i prodotti finora elencati sono ancora molto elementari con tecniche d'insegnamento non "ottimali". Molti di loro sono anche poco soddisfacenti a livello di qualità concettuale, eccetto i programmi della Xanadu (ditta che commercializza il software sviluppato nell'Università di Gerusalemme).
Adesso vediamo di rispondere a tipiche domande sui programmi CAI per musica:

- Cos'è e come funziona un programma CAI ?
- Cosa si può imparare con i programmi CAI ?

- I programmi CAI necessitano di una configurazione hardware speciale?
- Si applica l'Intelligenza Artificiale al CAI?

Cerchiamo di rispondere a tutte le domande in modo sistematico.

Cos'è e come funziona un programma CAI?
A livello generale un programma CAI è un programma di semplice uso (*friendly*) e interattivo, disegnato per insegnare una qualsiasi materia a un'utente standard (che diventa l'allievo).
A seconda delle tecniche d'implementazione, questi programmi si comportano in modo diverso; le tecniche più applicate in CAI sono:

- *Drill and Practice (Esercitazione):* l'obiettivo di questa tecnica è fare acquisire all'allievo delle abilità, durante l'esercitazione l'elaboratore propone degli esercizi i cui risultati possono essere memorizzati dall'elaboratore stesso.
  Dall'analisi dei risultati delle esercitazioni svolte e in base al percorso seguito dall'allievo per arrivare alle soluzioni, il programma CAI sceglie il grado di difficoltà delle esercitazioni successive.
  I programmi sono basati su esempi e su esercizi, durante i quali l'allievo riceve dei rinforzi e dei chiarimenti ulteriori, sia quando svolge correttamente l'esercitazione, sia quando commette degli errori.

- *Tutorial (istruzione)*: l'obiettivo di questa tecnica è fare acquisire all'allievo nuovi concetti; il programma (che esercita il ruolo di educatore) non solo espone esplicitamente dei concetti, ma guida l'allievo ad acquisire nuove conoscenze per mezzo di domande inerenti i contenuti trattati.

- *Instructional Game* (*gioco istruttivo*): l'obiettivo di questa tecnica è provare e stimolare le capacità di riconoscimento da parte dell'allievo dell'esistenza della "strategia" (in quel caso di gioco) e alla sua individuazione questi programmi permettono all'allievo di svolgere tanto il ruolo di avversario come quello di insegnante-istruttore.

- *Inquiry* (*indagine*): l'obiettivo di questa tecnica è stimolare lo sviluppo di capacità critiche da parte dall'allievo; il programma propone dei problemi, per risolvere i quali l'allievo deve trovare i metodi e i mezzi necessari, utilizzando solo le risorse dell'elaboratore.

- *Demonstration* (*dimostrazione*): l'obiettivo di questa tecnica è di stimolare la curiosità, il programma è fondamentalmente illustrativo e sviluppa presentazioni introduttive o specialistiche su una qualunque disciplina, permettendo all'allievo di manipolare l'ordine ed anche qualche parametro di queste presentazioni in modo che attraverso le proprie azioni percorra (impari) tutte le informazioni contenute nel programma.

- *Simulation* (*simulazione*): l'obiettivo di questa tecnica è favorire lo sviluppo dell'intuizione e fornire all'allievo una "macchina per esperimenti"; nei programmi di simulazione l'elaboratore si comporta come un dato sistema fisico, permettendo all'allievo di esercitare su di esso delle azioni, l'elaboratore mostra sul video gli effetti di tali azioni "simulando" il comportamento del sistema effettivo.

- *Problem Solving* (*soluzione di problemi*): l'obiettivo di questa tecnica è fare acquisire all'allievo un insieme di conoscenze strutturate unitamente alla capacità di analizzare e risolvere problemi. Il programma CAI richiede all'allievo la costruzione di una soluzione del problema proposto, soluzione che l'elaboratore poi è in grado di riconoscere.

Cosa si può imparare con i programmi CAI ?

A livello specifico dell'insegnamento musicale vediamo l'applicazione di queste tecniche CAI alle materie tradizionali dell'insegnamento musicale:

- *Drill and Practice* (*esercitazione*): è la tecnica appropriata per insegnare a "solfeggiare", "cantare" e "suonare" strumenti ovvero è la tecnica più interessante per favorire lo sviluppo della manualità musicale.

- *Tutorial* (*istruzione*): è la tecnica appropriata per insegnare la teoria della musica.

- *Inquiry* (*indagine*): è la tecnica appropriata per insegnare musicologia.

- *Demonstration* (*dimostrazione*): è la tecnica appropriata per insegnare la storia della musica.

- *Simulation* (*simulazione*): è la tecnica appropriata per insegnare l'acustica e la manipolazione del suono.

- *Problem Solving* (*soluzione di problemi*): è la tecnica appropriata per insegnare l'armonia il contrappunto e la fuga così come altre tecniche di composizione.

- *Instructional Game* (*gioco istruttivo*) è ovvio che è un'approccio senza limitazioni, nella misura in cui la musica tante volte nell'insieme dell'educazione scolastica è solo un gioco istruttivo.

I programmi CAI per musica richiedono una configurazione hardware speciale?

Si, però non per tutte le applicazioni, ad esempio necessitano di periferiche speciali nel caso in cui debbano controllare l'interpretazione dell'allievo su uno strumento.
Si potrebbe parlare delle potenziali architetture di sistemi per l'insegnamento musicale aiutato dall'elaboratore.
Oggi l'uso dello standard MIDI permette, in modo economico, di collegare periferiche, principalmente tastiere musicali ed ultimamente anche chitarre; è attraverso la MIDI che si può fare *Drill and Practice*, ovvero l'elaboratore è in grado di proporre degli esercizi musicali che l'allievo suonerà con le tastiere o la chitarra MIDI dopodiché l'elaboratore riconoscerà gli sbagli e proporrà la loro correzione in modo interattivo.
Per gli strumenti a fiato e per la voce è necessaria un'altra periferica non MIDI consistente in un circuito hardware di riconoscimento d'intonazione collegato da una parte all'elaboratore e dall'altra ad un microfono standard: l'allievo canta o suona lo strumento davanti al microfono che spedisce il segnale all'interfaccia. Da quel momento il programma CAI opera come sopraesposto.
In un futuro prossimo compariranno nel mercato altre periferiche MIDI capaci di trasmettere l'intonazione (*Pitch Recognition*). Il che permetterà alla MIDI l'uso di programmi *Drill and Practice* anche per qualsiasi strumento o voce.
Per il resto delle applicazioni CAI è necessario l'uso esclusivo dell'elaboratore fornito di possibilità grafiche evocative, interfaccia utente "amichevole" e grande disponibilità di memoria.

Si applica l'Intelligenza Artificiale al CAI?

I sistemi intelligenti per insegnare fanno capo alle ricerche sull'Intelligenza Artificiale o sui sistemi esperti.
Un sistema esperto è un ambiente specializzato per applicazioni. Ha come

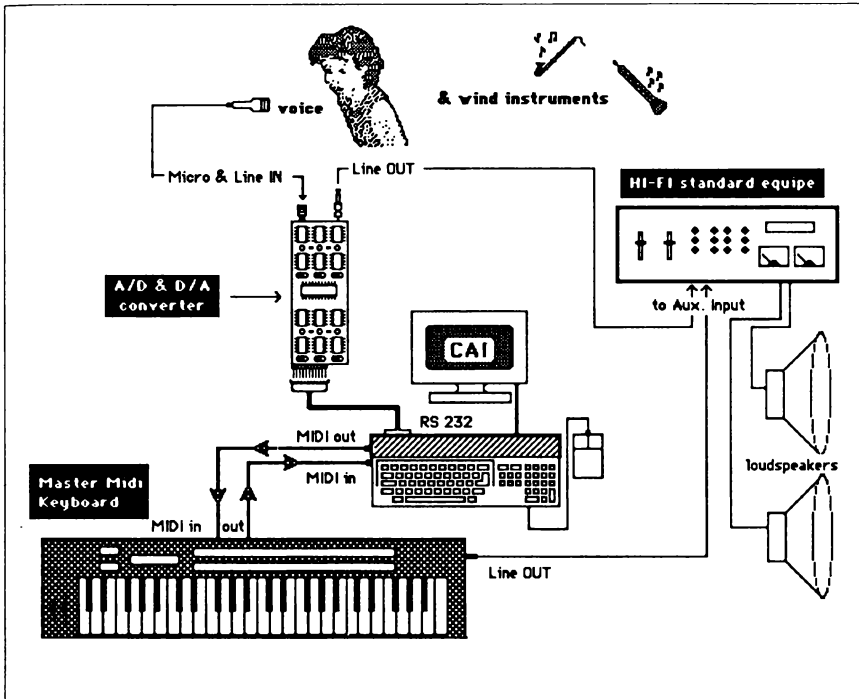



Fig. 1.41 Architettura hardware standard (a basso costo) per applicazioni CAI (*Computer Aided Instruction*) musicali.

caratteristica fondamentale il fatto che "impara" dalla sua esperienza accumulata in una base di dati che viene chiamata "base di conoscenza".
Questi sistemi applicati all'insegnamento generano un nuovo settore denominato ICAI (o *Intelligent* CAI) e la differenza con il CAI tradizionale consiste nella gestione del dialogo allievo-elaboratore.
Nell'ICAI anche l'allievo può prendere l'iniziativa, può rispondere a un quesito con un altro quesito, l'interazione allievo-elaboratore risulta così molto simile a un vero dialogo a iniziativa mista.
Non esistono sistemi esperti con applicazioni musicali a livello popolare, ma esistono solo a livello sperimentale (in Italia esiste un solo sistema esperto applicato alla musica nel DIST dell'università di Genova); il costo di questo tipo di apparecchiatura (hardware e software) è oggi estremamente elevato.

Il progetto LOGO

il LOGO è un linguaggio di programmazione molto semplice sviluppato dal laboratorio per la ricerca sull'Intelligenza Artificiale presso il MIT (Massachusetts Institute of Technology) che serve "a insegnare a pensare" in un'ottica di sperimentazione sia a bambini che ad adulti non specializzati in informatica. Questo linguaggio aiuta a sviluppare negli allievi che lo utilizzano la capacità d'imparare da soli, riflettere sulle conoscenze apprese e sulle procedure che conducono a risolvere dei problemi.
Sono disponibili varie visioni del LOGO tra le quali una per Atari ST.

Difficoltà nella realizzazione e nella divulgazione dei programmi CAI

Lo sviluppo di programmi CAI per musica è un lavoro costoso (probabilmente sono i programmi più costosi) e difficile perché è estremamente inter-disciplinare, compromette studiosi e tecnici in materie cosi differenti come didattica, psicologia, musicologia, informatica, ingegneria elettronica, fisica acustica, psicoacustica ecc.
La sua implementazione e la sua accettazione coinvolge concetti che vanno dalla teoria della percezione e l'ottimizzazione dei processi psico-motori (manualità) fino ad altri meno teorici e più terra terra come problemi commerciali e burocratici.
Qualsiasi nuovo prodotto per l'insegnamento, prima di essere assunto come tale nelle scuole pubbliche, ha bisogno di un'approvazione tecnico-politica da parte degli Enti Statali preposti alla Pubblica Istruzione.

Atari ST e CAI

I computer Atari ST permettono perfettamente di lavorare nelle applicazioni CAI perché hanno le caratteristiche appropriate per questo tipo di applicazioni: interfaccia utente amichevole (mouse, desktop), grafica evoluta (GEM), interfaccia MIDI standard nelle configurazioni di base, chip sonoro interno, memoria da 512 Kb a 4 Mb in RAM, hard-disk collegabile, prezzo contenuto. Dunque le prospettive del loro uso nell'insegnamento musicale sono enormi. Infatti sono presenti in tutti i centri di ricerca e sviluppo software musicale tanto in Italia quanto nel resto dell'Europa e negli USA.

2

MIDI e dintorni

La musica è la scienza dei numeri riferita ai suoni.

J. de Garlande, sec. XIII

Quando nel gennaio del 1983, negli Stati Uniti, in una sessione di lavoro della National Association of Music Merchants (dove erano presenti i più importanti costruttori di strumenti musicali del mondo) Dave Smith e Chet Wood, due informatici musicali, collegarono con due cavetti DIN due strumenti di due fabbricanti diversi, fino a quel momento assolutamente incompatibili fra di loro, e li fecero andare "insieme", prima uno schiavo dell'altro poi alla rovescia e infine tutti e due controllati da un elaboratore, qualcosa di rivoluzionario si era prodotto nella storia della musica "popolare" della fine del secolo XX: l'apparizione della MIDI.
Dave e Chet avevano realizzato quest'impresa solo mediante una piccola variazione fatta incorporando un po' di "chip" in ogni strumento, e soprattutto con l'uso di un "codice di comunicazione" proposto da loro e diventato dopo quella piccola prova, senza alcuna formalità né cerimonia speciale, il *primo standard mondiale di comunicazione dell'informazione musicale digitalizzata.*
La prima volta che Dave e Chet fecero conoscere la loro proposta fu nella settantesima Convention della Audio Engineering Society nel novembre del

1981: in quel momento chiamarono il loro codice "Universal Synthesizer Interface".
Con piccole variazioni presentarono ufficialmente la versione definitiva nel dicembre del 1982, definendola MIDI, sigla che sta per *Musical Instrument Digital Interface.*
La MIDI fu adottata ufficialmente dai fabbricanti di strumenti musicali tanto americani come giapponesi nel febbraio del 1983. Dunque è passato pochissimo tempo fisico, però la MIDI ha già creato una nuova generazione di musicisti, perchè esiste tutta una "offerta" da parte dei fabbricanti di apparecchi con la MIDI. Questa è la vera potenzialità degli "standard".

2.1. Tutto quello che avresti voluto sapere della MIDI.. ma non hai mai osato chiedere.

Probabilmente Woody Allen non girerà mai un film con questo titolo (sicuramente il sesso è più popolare della MIDI), però anche in questo campo non mancano le confusioni e le ambiguità, qualcuna divertente, altre meno.
La prima cosa che puoi fare, se vuoi sapere sulla MIDI qualcosa che non trovi in questo libro, è scrivere in California. Lì c'è la

International MIDI Association
11857 Hartsook Street - North Hollywood -CA 91607
tel. (818) 505 89 64

che pubblica mensilmente un bollettino sulle novità "MIDI".
Generalmente lì ai privati non danno molto retta, perché si tratta fondamentalmente di un centro di lavoro della MIDI Manufacturers Association, comunque forniscono un documento (contro pagamento, è chiaro) che si chiama *MIDI 1.0 Detailed Specification*, che è il documento "ufficiale" della codifica MIDI.
A proposito, puoi anche risparmiare i soldi perché in questo capitolo ho cercato di tradurlo dall'inglese per te.
Ci sono altri servizi più "orientati all'utente finale" che non le Associazioni MIDI:

Club MIDI Software
P.O. Box 93895 - Hollywood -CA 90093

è una software house collegata all'associazione.

Se vuoi spendere in telefonate intercontinentali ci sono due "reti" di telecomunicazione via MODEM:

MIDI Connection (212) 594 -2646 (telecoms)

e

MIDI World Network (213) 826 42 88 (telecoms)
11920W Olympic Blvd - Los Angeles- CA 90064

Prima di spiegare in termini più tecnici la codifica MIDI, è bene introdurre un po' di concetti chiari e semplici.

Cos' è la MIDI?
Semplicemente un protocollo di comunicazione tra strumenti digitali. Ovvero un "codice" di comunicazione fra di essi.

Perché un apparecchio è MIDI e un altro no?
Gli apparecchi che sono MIDI...zzati, sono quelli preparati ad accettare e dunque a capire la codifica MIDI.
Quelli non-MIDI semplicemente non l'hanno implementata.

Esistono apparecchi MIDI, che non accettano tutta la codifica MIDI?
Sì, tantissimi; l'implementazione industriale della MIDI si è andata realizzando in modo progressivo, ci sono apparecchi che accettano di essere controllati via MIDI solo in qualcuno dei loro parametri e in altri no.
Per evitare problemi, tutti i fabbricanti forniscono nei manuali di uso "il livello d'implementazione della MIDI" dell'apparecchio in questione.

Si possono MIDI...zzare apparecchi non MIDI?
Dipende..., il problema è complesso. Se l'apparecchio è un sintetizzatore o una "drum machine" della generazione anteriore alla MIDI (tipo Moog, Korg, Roland ecc) predisposto ai protocolli di comunicazione particolari (CV, Trigger ecc.), si può effettuare la MIDI...zzazione in modo molto parziale (mai totalmente MIDI) usando interfacce speciali che sono convertitori di protocollo, per esempio MIDI-CV ecc.
Normalmente non ne vale la pena, innanzitutto per il prezzo (elevato), poi per la mancanza di affidabilità (ritardi e desincronizzazioni non tollerabili a livello professionale).

Gli strumenti non digitali *non si possono MIDIzzare*: ad esempio, le chitarre elettriche non accettano la MIDI perché sono strumenti elettroacustici analogici. Le chitarre veramente MIDI (Syntaxe, Stepp, Casio ecc) sono strumenti tecnologicamente complessi, simili alle chitarre elettriche solo nella forma. Esistono box-effetti che, con qualsiasi suono analogico, permettono di controllare qualche parametro di un apparecchio MIDI, partendo da un ingresso micro/linea, mediante l'uso di algoritmi di trasformazione (filtri fondamentalmente). La trasformazione non è lineare, cioè non è fedele ai parametri musicali in ingresso; questi box-effetti vanno intesi come "pedali effetti" e mai come una MIDIzzazione di una chitarra o di un basso perché in realtà non sono capaci di comunicare quello che è l'essenziale nella MIDI — la trasmissione di *tutte* le informazioni di controllo musicalmente significative.

Perché un computer ha la MIDI e un altro no?
Per cominciare, tutti i computer che hanno la porta seriale di comunicazione RS 232 possono accettare l'implementazione *totale* della MIDI.
L'implementazione MIDI, in un computer che non possiede questa interfaccia, richiede due cose: "un po' di hardware" e "un po' di software".
Il circuito che controlla la MIDI, il "po' di hardware" che non hanno i computer di serie e che bisogna aggiungere "in scheda", si chiama UART ovvero *Universal Asynchronous Receiver Transmitter*. Questa UART ha delle caratteristiche particolari di gestione e dunque ha bisogno di un "po' di software", ovvero di riscrivere i "driver" che permettono il suo controllo attraverso la porta seriale RS232.
Normalmente le schede di interfaccia (tra porta seriale RS232 computer e MIDI) hanno un prezzo che oscilla tra i 200 e i 400 dollari (cioè fra le 300 e le 600 mila lire) e la più parte dei loro fabbricanti non forniscono il codice sorgente dei loro driver, il che rende difficile la possibilità di sviluppo di software che coinvolga in qualche modo l'uso di queste interfacce da parte degli utenti avanzati a livello informatico.
Gli Atari ST hanno la MIDI implementata nei modelli "di serie", dunque non hanno bisogno di una scheda supplementare di interfaccia e i sorgenti dei loro driver sono di dominio pubblico.

Queste sono risposte a domande che ho ricevuto normalmente da musicisti che lavoravano con me in studio di registrazione nel corso di questi ultimi anni, in cui la codifica "si stava facendo strada". Ho voluto riproporle perché presentano dubbi abbastanza generali che bisogna chiarire prima d'iniziare a spiegare "Come è e che cosa controlla la codifica MIDI".

Di che natura è la codifica MIDI?
Digitale, ovvero numerica, la codifica viene organizzata in base a numeri. Ogni numero significa una cosa diversa.

Che cosa può controllare la MIDI?
La MIDI può controllare molte cose, non solo quelle che rappresentano il nome dei suoi dati "tipici", ma anche attraverso questi stessi messaggi tutto quanto l'immaginazione degli sviluppatori hard & soft sarà capace di volerle fare controllare in futuro. Ad esempio il messaggio *key-on* in una periferica diversa da una tastiera potrebbe servire a comandare l'accensione di un qualsiasi interruttore (così si possono controllare le luci di una discoteca con la MIDI).

2.2. Hardware e specifiche della MIDI

L'interfaccia MIDI, essendo implementata in tecnica digitale (dall'inglese *digit* = numero), lavora esclusivamente su numeri che rappresentano l'evento musicale. Il perché non lavora in analogico è molto semplice: se una grandezza digitale assume valori identificabili con precisione, analogicamente avremo invece grandezze variabili con continuità, ad esempio fra un massimo e un minimo. Cio che è possibile rilevare, in questo caso, sarà solo il cambiamento di stato, ma non i valori intermedi che sono stati assunti, cioè riceviamo "una" sola informazione.
Digitalmente questo non avviene, poiché usando numeri che verrano poi decodificati dai circuiti interni dei microprocessori, potremo ricevere "più" informazioni; le quali, nel caso specifico della MIDI, sono valori compresi fra 0 e 255 (campo o *range* di variabilità).
Il primo passo per la trasmissione è per ogni numero la conversione in forma binaria, poiché è più significativa per la rappresentazione del segnale elettrico digitale: 0 quando il circuito elettronico è aperto e 1 quando è chiuso. Ovvero parliamo di bit, i quali se usati in gruppi di 8 costituiscono il byte, il byte permette la rappresentazione in binario di tutti i numeri fra 0 e 255 (2^8).
L'interfaccia digitale MIDI trasmette un byte alla volta con un formato di dieci bit, di cui solo 8 rappresentano i parametri che stiamo usando; gli altri due sono bit di *start* (inizio) e di *stop* (d'arresto).
In MIDI è prevista solo la "trasmissione seriale delle informazioni", ovvero in sequenza (una dietro l'altra), su una sola linea, il che potrebbe causare una minore velocità di trasmissione, offrendo però in contropartita collegamenti economici e semplici.

Il circuito di trasmissione MIDI (MIDI out) consiste in una UART, ovvero di un *Universal Asynchronous Receiver Trasmitter.*
L'interfaccia opera mediante l'invio e la ricezione di un flusso asincrono di dati seriali a una frequenza di 31.25 Kbaud (clock di 1 MHz diviso per 32) con un bit di start, otto bit significativi ed uno di stop, in totale 10 bit che sono trasmessi in 320 millisecondi ovvero 3125 bit o caratteri al secondo.
Il mezzo di supporto alla propagazione dei segnali è un semplice loop di corrente da 5mA (lo "0" logico corrisponde a corrente ON) optoisolato con un unico connettore da 5 pin del quale vengono utilizzati solo 3 pin.
Dunque i connettori sono DIN-femmina a 5 poli disposti a 180 gradi, ovvero lo standard dei connettori degli impianti hi-fi.
Il cavo di connessione tra unità dialoganti è formato da una coppia di fili intrecciati e schermati, con lo schermo a massa solo al terminale sorgente, (quest'ultimo è indispensabile per evitare un possibile disturbo a 50 Hz dovuto a un'eventuale differenza di potenziale tra le due terre). Ogni coppia di questi fili costituisce una linea di trasmissione unidirezionale.
I dispositivi MIDI sono dotati di 3 prese: una uscita MIDI-Thru che è la copia, riprodotta fedelmente, del segnale presente nel MIDI-In (ingresso per la

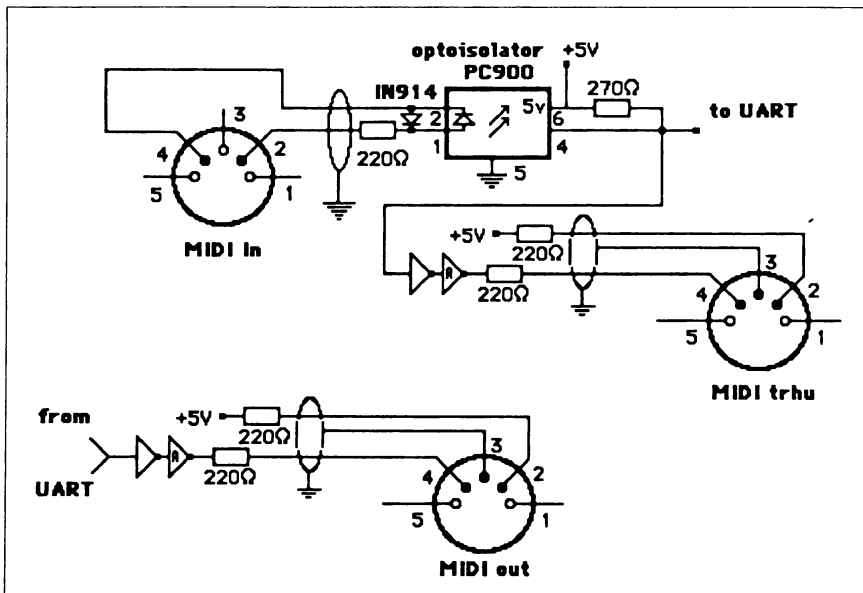



Fig. 2.1 I connettori MIDI.

PROTOCOLLO MIDI DATI DI CANALE			
Range	Dato o messaggio	Byte di stato	Byte di dato
144-159	Nota accesa (key-on) (key-on velocity)	1 0 0 1 b b b b nota ON	0 k k k k k k k Numerazione dei tasti (0-127) 0 v v v v v v v Velocità di discesa del tasto
128-143	Nota spenta (key-off) (key-off velocity)	1 0 0 0 b b b b nota OFF	0 k k k k k k k Numerazione dei tasti (0-127) 0 v v v v v v v Velocità di rilascio del tasto
160-175	Variazione pressione tasto (polyphonic key pressure)	1 0 1 0 b b b b	0 k k k k k k k Numerazione dei tasti (0-127) 0 v v v v v v v Pressione esercitata sul tasto
176-191	Variazione posizione di controllo (control change)	1 0 1 1 b b b b	0 c c c c c c c Indirizzo del controllo (0-121) 0 v v v v v v v Valore in uscita del controllo
192-207	Cambio di programma (program change)	1 1 0 0 b b b b	0 n n n n n n n Numero del nuovo programma selezionato (0-127)
208-223	Pressione del tasto assegnato al canale (channel pressure)	1 1 0 1	0 v v v v v v v Valore di controllo variabile con continuità (7 bit di risoluzione)
224-239	Pitch bend (glissando)	1 1 1 0	0 L L L L L L L 0 h h h h h h h Valori di controllo variabili con continuità (14 bit di risoluzione)
122-127	Channel Mode messaggi: Controllo locale spento Controllo locale acceso Tutte le note spente Omni mode OFF (spento) Omni mode ON (acceso) Mono mode ON Poly mode ON	 1 0 1 1 b b b b 1 0 1 1 b b b b 1 0 1 1 b b b b 1 0 1 1 b b b b 1 0 1 1 b b b b 1 0 1 1 b b b b 1 0 1 1 b b b b	 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 z z z z z z z (z = numero dei canali) 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

Tab. 2.1 I messaggi di canale

ricezione dei dati), e la terza MIDI-Out per i dati in trasmissione (prodotti per la UART di quella periferica).
Il cavo di collegamento, che permette di usufruire di 16 canali, sappiamo già che è schermato opportunamente per evitare eventuali disturbi, sebbene le interferenze aumentino proporzionalmente alla lunghezza dei fili.
Esistono due tipi di circuito: quello di interfaccia, che adatta i dati seriali in

uscita dall'UART al segnale elettrico del cavo MIDI, e quello per l'interfacciamento seriale, che controlla lo scambio con altri dispositivi collegati alla MIDI.

2.3. Il codice MIDI

Ogni informazione in MIDI possiede una propria codifica realizzata con due tipi di byte: quelli di stato e quelli di dati.
Byte di stato: sono chiamati così perché specificano lo stato del sistema ovvero il parametro su cui agire (nota accesa e spenta, parametri di pressione, cambiamenti di programmi ecc.). Il loro bit più significativo è sempre 1.
Byte di dati: sono significativi riguardo al valore da controllare; si riferiscono, per esempio, alla numerazione dei tasti della tastiera utilizzata, e dei programmi di cui lo strumento è dotato, o alla posizione (indirizzo) di un dato controllo. Il primo bit, lo 0, è quello più significativo.
Un messaggio MIDI consiste di un singolo byte di stato seguito da 0, 1 o 2 byte di dati quantizzati secondo le specifiche dell'interfaccia. I tipi di informazioni previsti nella codifica MIDI sono due: dati di canale e dati di sistema.

2.3.1. *Dati di canale e di sistema. Trasferimento delle informazioni.*

Abbiamo visto che un byte di stato è nella forma:

```
1 a a a b b b b
```

dove `1` è il tipico bit di stato e `b b b b` specifica il messaggio.
Se `a a a` non è `111` parleremo di *dati di canale*, ovvero di dati disponibili per essere trasmessi su uno dei 16 canali; se invece `a a a` è proprio `111` avremo i cosiddetti *dati di sistema* che analizzeremo più avanti.
Si parlerà di dati di canale qualora si abbiano informazioni che comunicano quali note suonare e con quale durata. Ad esempio, `144` è il codice che corrisponde a "nota accesa", mentre `128` corrisponde a "nota spenta".
Nella tabella riassuntiva si vedono i valori numerici che corrispondono agli specifici byte di stato e di dati e le funzioni che controllano.
È da notare che la distinzione fra dati di canale e di sistema avviene perché alcuni messaggi devono essere ricevuti da tutti gli strumenti (questi riguardano lo stato) mentre altri devono rimanere confinati in uno dei 16 canali disponibili

(e sono appunto i dati di canale). I canali in MIDI sono indirizzi e non entità fisiche, e sono contenuti nei byte di stato.

2.3.1.1. *Messaggi di canale*

Vediamo cosa e come controllano i messaggi di tipo canale:

Key on (nota accesa) e Key-on velocity (velocità di tasto)

Questo messaggio si verifica ogni volta che un tasto della tastiera musicale viene premuto. Quando verrà rilasciato avremo il dato di nota - off.
Vediamo la codifica:

```
byte di stato =    1 0 0 1 b b b b
byte di dati  =    0 k k k k k k k
                   0 v v v v v v v
```

dove le cifre `kkkkkkk` specificano quale nota è stata premuta e le cifre `vvvvvvv` parametrizzano la velocità di discesa del tasto con tempi varianti fra 40 e 5 millisecondi. Entrambi i valori saranno compresi fra `0` e `127`, il che dà un ventaglio di note possibili uguale a `127` note (dieci ottave e mezza), con il valore `60` assegnato al Do centrale della tastiera (C3)
La velocità di discesa del tasto viene equiparata alla dinamica (volume della nota) che in termini musicali permette una gamma dal più che pianissimo al più che fortissimo (con 126 punti di graduazione intermedia).

NB la velocità uguale a 0 viene interpretata come spegnimento della nota.

Key-off (nota spenta) e Key-off velocity (velocità di rilascio del tasto)

Come i messaggi relativi alla nota accesa (key-on e key-on velocity) opera il controllo della nota spenta (key-off e key-off velocity). La sua codifica è:

```
byte di stato = 1 0 0 0 b b b b
byte di dati  = 0 k k k k k k k
                0 v v v v v v v
```

dove le cifre kkkkkkk specificano quale nota è stata spenta e le cifre vvvvvvv parametrizzano la velocità di rilascio del tasto con tempi varianti fra 40 e 5 millisecondi.
Comunque questo parametro è implementato solo in pochissimi apparecchi e il suo uso è poco frequente. In genere servirebbe per associare al rilascio del tasto un effetto potenzialmente di sustain o "coda del suono". A rilascio veloce corrisponde poca coda: effetto di ammazzare il suono. A rilascio lento corrisponde molta coda: effetto di lasciare per lungo tempo la "risonanza" del suono.

Polyphonic key pressure (variazione di pressione sul tasto)

Indica la variazione di pressione che si esercita sul tasto del sintetizzatore. È bene ricordare che la variazione di pressione sul tasto in molti sintetizzatori può generare un effetto tipo vibrato, simile alla tecnica del violino o del sax ecc., ovvero: prima la nota precisa e dopo un vibrato sulla coda, oppure l'effetto contrario (prima vibrato dopo no).
Il formato è:

byte di stato = 1 0 1 0 b b b b
byte di dati = 0 k k k k k k k
0 v v v v v v v

dove le cifre kkkkkkk specificano quale nota o tasto è stata premuta e le cifre vvvvvvv parametrizzano la forza esercitata sul tasto nell'istante in cui il dato di canale è stato trasmesso.
Poiché tale forza varia con continuità dovranno essere trasmessi più dati al secondo.

NB In linea di principio qui c'é un dubbio importante nella comprensione della codifica, perché questo parametro dovrebbe variare con continuità, ma le specifiche MIDI non dicono nulla sulla frequenza di campionamento del parametro (e conseguentemente sulla frequenza di emissione dati); il costruttore si riserva quindi la facoltà di scegliere la frequenza più opportuna, che presumibilmente è funzione delle prestazioni globali dei dispositivi di trasduzione.

Control change (variazione delle posizioni di controllo)

Specifica una variazione dei parametri operativi all'interno del sistema.
Il formato è:

byte di stato = 1 0 1 1 b b b b
byte di dati = 0 c c c c c c c
0 v v v v v v v

dove ccccccc (0 - 121) contengono l'indirizzo numerico del controllo (questi indirizzi sono specificati nella tabella associata di funzioni, ove i valori da 98 a 121 sono riservati per assegnazioni future) e vvvvvvv è il nuovo valore di uscita del controllo cambiato.
In questo ambito potremo vedere tre casi distinti:

1° caso: Il controllo riguarda semplici "switch" (cambi da una funzione ad un' altra) dove vvvvvvv può assumere i valori previsti da 0 a 127 nella tabella di funzioni.
2° caso: Il controllo riguarda elementi che variano con continuità ed è sufficiente una risoluzione pari a 128.
3° caso: È necessaria una risoluzione più spinta; per questa situazione i numeri corrispondenti da 0 a 31 di ccccccc sono indirizzi per i 7 bit più significativi di 32 controller differenti, i numeri da 32 a 63 di ccccccc sono indirizzi per i 7 bit meno significativi degli stessi controller; cosi si ottiene una risoluzione di una parte su 21^{4}.

Program change (cambio di programma)

Un programma a livello di questa codifica non viene interpretato nel senso informatico (software) ma solo come un "insieme completo di parametri", ovvero definiamo "programma" l'insieme globale dei valori assunti dai parametri che governano il funzionamento di una macchina in un certo istante. Ad esempio, ogni suono in un DX7 viene inteso come un programma, una determinata organizzazione di campioni in un campionatore è un programma, un determinato insieme di posizioni dei "faders" in un mixer MIDI è un programma, ugualmente negli apparecchi di effetti MIDI (riverbero, eco, delay, ecc).

Il formato è:

byte di stato = `1 1 0 0 b b b b`
byte di dati = `0 n n n n n n n`

dove `nnnnnnn` è l'indirizzo del nuovo programma e può variare fra `0` e `127`, dunque si possono controllare 127 programmi diversi.

Channel pressure (pressione del tasto assegnato al canale)

Questa informazione, associata a un dato di canale, produrrà un segnale di controllo che varia con continuità (a differenza del *polyphonic key pressure* visto in precedenza è un segnale di controllo globale associato a un canale). Il formato è:

byte di stato = `1 1 0 1 b b b b`
byte di dati = `0 v v v v v v v`

dove `vvvvvvv` specificherà il nuovo valore del parametro.
Il *channel pressure* in genere è implementato dalle tastiere con un solo sensore di pressione; si rileva quindi la pressione globale esercitata sui tasti. Può fornire rapide variazioni del suono, considerato globalmente a bassa risoluzione.

Pitch bend (glissando)

È sempre un messaggio che segnala una variazione con continuità associata a un canale.
Il formato è:

byte di stato = `1 1 1 0 b b b b`
byte di dati = `0 L L L L L L L`
`0 h h h h h h h`

Usualmente tale dato proviene dall'uso di una rotella o di un joystick nelle tastiere e da sofisticate implementazioni tastiera-corde nelle master MIDI guitar.

Il suo effetto più conosciuto è la variazione dell'intonazione con continuità. La sua definizione è di 14 bit (`7 L + 7 h`), il che permette un livello d'informazioni molto superiore al resto dei dati MIDI.
Il numero di dati diversi possibili è 2^{14} = 16.384 posizioni di controllo (sufficienti per simulare la continuità analogica).

Channel Mode (modi di operare del canale)

Queste informazioni non controllano parametri di generazione sonora, bensì specificano i modi in cui operano i vari canali (Omni, Poly, Mono).
Il formato è:

```
byte di stato  =   1 0 1 1 b b b b
byte di dati   =   0 c c c c c c c
                   0 v v v v v v v
```

Nella terminologia MIDI i termini Omni, Poly e Mono indicano i modi di assegnamento voci-canali:

- *In modo Omni*:
  lo strumento riceve messaggi Channel voice su tutti i canali, ignorando quindi i 4 bit meno significativi del byte di stato.

- *In modo Poly*:
  la macchina risponde solo sul canale sul quale è stata predisposta.

- *In modo Mono*:
  ogni generatore interno o voce viene assegnato ad un canale, lo strumento si comporta in modo multitimbrico. (I sintetizzatori possiedono da 1 a *n* generatori che determinano il grado di polifonia della macchina: il modo Mono consente di gestire indipendentemente i generatori.)

Le modalità di assegnazione dei modi sono rappresentate nella tabella dei messaggi di canale.
Prefissato un canale base, sulla macchina si possono avere 4 combinazioni, rispettivamente Omni-on ed Omni-off in Poly e Mono.

	Modi	Descrizione
1	OMNI ON - POLY	Messaggi ricevuti su tutti i canali ma trasmessi solo sul canale base detto n.
2	OMNI ON - MONO	Messaggi ricevuti su tutti i canali, assegnati uno alla volta per il singolo controllo, trasmessi sul canale n
3	OMNI OFF - POLY	Messaggi ricevuti e trasmessi solo sul canale base n.
4	OMNI OFF - MONO	Messaggi ricevuti e trasmessi solo su un set di canali a partire da quello base.

Tabella 2.2 I modi della MIDI.

Running Status

Un'eventualità ricorrente nella trasmissione dati è quella in cui molti dati contigui possiedono lo stesso byte di stato (ad esempio l'accensione simultanea di diverse note).
Per ovviare a problemi di velocità di trasmissione (la trasmissione di un byte, ricordiamo, occupa un tempo pari a 320 μsec.) esiste una direttiva (alla quale tutti i costruttori si devono adeguare) secondo la quale i diversi byte di dati contigui vengono associati all'ultimo byte di stato ricevuto.
Gli strumenti devono quindi possedere il necessario supporto hardware per "ricordare" l'ultimo byte di stato ricevuto.

2.3.1.2. *Messaggi di sistema*

Sono informazioni rivolte a tutte le unità asservite, non contengono quindi il campo d'indicazione del canale.
Si suddividono in 3 tipi e sono caratterizzati dall'avere i 4 bit più significativi del byte di stato uguali a uno.

1° tipo: sistema tempo reale

I messaggi del 1° tipo sono utilizzati per sincronizzare unità distinte in tempo reale oppure in tempi differiti; la procedura tipica per farlo consiste nella registrazione su nastro di una traccia di sincronismo e nell'effettuare poi registrazioni successive mediante tecnica multitraccia.
Sono messaggi che possono essere trasmessi con una periodicità abbastanza rapida, quindi permettono di sincronizzare ad esempio un sequencer con gli

Funzioni	Codice
Modulation wheel	1
Breath controller	2
Foot controller	4
Portamento Time	5
Data entry	6
Main volume	7
Sustain pedale	64
Portamento	65
Sostenuto	66
Soft pedal	67
Data + increment	96
Data – decrement	97

Tab. 2.3 Funzioni e indirizzi della MIDI.

altri elementi del *network* (la rete, concetto che chiariremo più avanti).
In esso si distinguono:

- Impulso Syncron o *timing clock*: in cui il segnale è trasmesso 24 volte ogni quarto di misura;
- Da capo o *start*, per partire all'inizio della sequenza;
- *Continue*: per ripartire da un punto continuando la sequenza;
- *Stop*: per fermare;
- *Active sensing*: trasmesso tre volte al secondo indica l'attività di trasmissione in atto;
- *Reset*: per reinizializzare l'intero sistema alle condizioni di power-on.

2° tipo: dati di sistema comune

Sono comandi per tutte le unità collegate, quali sequencer, sintetizzatori, expander ecc., e riguardano il numero della misura, il codice del brano musicale e anche l'accordatura degli oscillatori.
Vediamo i messaggi:

- Numero del brano (*song select*), permette di selezionare un brano all'interno di una memoria di 128 sequenze. Il formato è:

byte di stato = 1 1 1 1 0 0 1 1
byte di dati = 0 $ $ $ $ $ $ $

PROTOCOLLO MIDI DATI DI SISTEMA			
Codice	**Messaggio**	**Byte di stato**	**Byte di dato**
248	Impulso Synchron (timing clock)	1 1 1 1 1 0 0 0	
250	Da capo (start)	1 1 1 1 1 0 1 0	
251	Continue	1 1 1 1 1 0 1 1	SISTEMA TEMPO REALE:
252	Stop	1 1 1 1 1 1 0 0	FORMATO:
253,254	Active sensing	1 1 1 1 1 1 1 0	(1 1 1 1 1 x x x)
255	Reset	1 1 1 1 1 1 1 1	
242	Numero della misura	1 1 1 1 0 0 1 0	0 L L L L L L L 0 h h h h h h h 14 bit; per la numerazione delle battute contate dall'inizio della canzone
			SISTEMA COMUNE:
243	Numero del brano	1 1 1 1 0 0 1 1	0 s s s s s s s FORMATO: 7 bit; per il (1 1 1 1 0 x x x) numero del brano
246	Richiesta di accordatura (Tune request)	1 1 1 1 1 1 1 0	
240	Sistema esclusivo (start)	1 1 1 1 0 0 0 0 inizio	
			0 a a a a a a a
		0 x x x x x x x corpo	SISTEMA ESCLUSIVO
			0 x x x x x x x
247	Sistema esclusivo (end)	1 1 1 1 0 1 1 1 fine	

Tab. 2.4 Messaggi di sistema MIDI.

- Numero della misura (*song position pointer*), facilita il posizionamento in un dato punto della sequenza in atto. Il formato è:

  byte di stato = `1 1 1 1 0 0 1 0`
  byte di dati = `0 j j j j j j j`
  `0 h h h h h h h`

  dove `jjjjjjj` e `hhhhhhh` specificano un numero a 14 bit (con un'estensione massima di 214 e questo essi fa specificando i battiti (6 colpi di clock MIDI) partendo dall'inizio della sequenza.

- Richiesta di accordatura (*tune request*) codifica le operazioni per accordare gli oscillatori della tastiera.

3° tipo: sistema esclusivo

Sono messaggi dedicati a un particolare tipo di macchina, consentono quindi

di entrare nel cuore di uno strumento e di manipolare i parametri che governano il suo funzionamento.
Ovvero trasmissioni specifiche riguardanti un dato dello strumento. Il loro formato è rappresentato nella tabella dei dati di sistema ove il primo byte di stato è per l'inizio del sistema esclusivo, mentre il secondo è l'identificazione propria di costruzione della MIDI Association (USA) o della Japan MIDI.
Ad esempio:

Sequential Circuits	01H
Big Briar	02H
Kawai	40H
Roland	41H
Korg	42H
Yamaha	43H

Le *xxxxxxx* sono per le informazioni specifiche, mentre l'ultimo byte è la fine della trasmissione in sistema esclusivo.
Nel sistema esclusivo il costruttore può definire personalmente quanti byte di stato devono essere usati, tutto il resto deve essere codificato come abbiamo già visto.
Dunque, controlli come VCF, VCA, DCO, Key transpose, LFO, Envelope, non sono codificati univocamente: ogni costruttore determinerà l'implementazione e poiché qualsiasi strumento ha le proprie caratteristiche non possiamo aspettarci lo stesso comportamento per tutti.
Per controlli nel range fra 176 e 191, sarà il costruttore a fornire la necessaria documentazione per chiarire la codifica adottata.
Solo con strumenti identici le codifiche saranno le stesse. In collegamenti con differenti marche potrebbero insorgere incongruenze.

2.3.2. *Informazione MIDI per il trasferimento di campioni*

Le specifiche MIDI definiscono infine un protocollo che permette di effettuare trasferimenti di campioni da una macchina all'altra.
Di norma i dati sono preceduti da una *intestazione* seguita da un certo numero di *pacchetti di dati*.
L'intestazione è costituita da:
1. Numero del campione (0 - 16383)
2. Formato del campione (8 - 28 bit) lineare

3. Periodo di campionamento (1-2.097.151 nanosecondi) e frequenza di campionamento da 1 GHz a 336 Hz
4. Lunghezza del campione (1-2.097.151 words)
5. Punti di loop
6. Tipo di loop

I *pacchetti di dati* sono così formati:
1. Numero del pacchetto;
2. 120 byte di dato in formato lineare;
3. Checksum.

Inoltre vengono utilizzati i seguenti messaggi:
1. Sample Dump Request
2. Cancel Dump
3. Not Acknowledge
4. Acknowledge

NB: Per le modalità di trasferimento dati è necessario fare riferimento al tipo di macchina utilizzata (bisogna quindi consultare sui manuali d'uso la parte che riguarda l'implementazione MIDI)

Visti fino qui tutti gli aspetti della codifica MIDI adesso vediamo altri aspetti hardware e di collegamento per l'uso degli apparecchi muniti di MIDI.

2.4. **Interfacciamento con il computer: architetture e limitazioni**

Il collegamento con il computer avviene tramite la porta seriale opportunamente adattata alle specifiche per l'interfaccia MIDI.
Gli Atari ST sono gli unici computer che portano questa interfaccia nei modelli di base senza bisogno di ulteriori circuiti.
Nel realizzare l'architettura fra strumenti via MIDI, lavorando in tempo reale, la rete deve essere realizzata in modo tale da rendere minimi i ritardi causati dalla lunghezza dei cavi onde evitare sfasamenti nell'esecuzione delle sequenze.

2.4.1. *Architetture MIDI*

Per quanto riguarda le architetture possiamo ottenere diverse configurazioni:

- *In parallelo* - catena di thru
  dove la Master MIDI controlla ai altri elementi che sono i suoi "schiavi".

- *In serie* - catena out-in
  dove il primo sintetizzatore controlla il secondo, il secondo il terzo e così via.

- *Ibride*
  dove la Master MIDI può controllare l'elaboratore e viceversa, e, comunque sia questo controllo, entrambi controllano le altre periferiche.

2.5. **Problemi e limitazioni del MIDI**

A parte il problema concettuale della serialità ci sono anche altri tipi di problemi a cui adesso accenneremo.

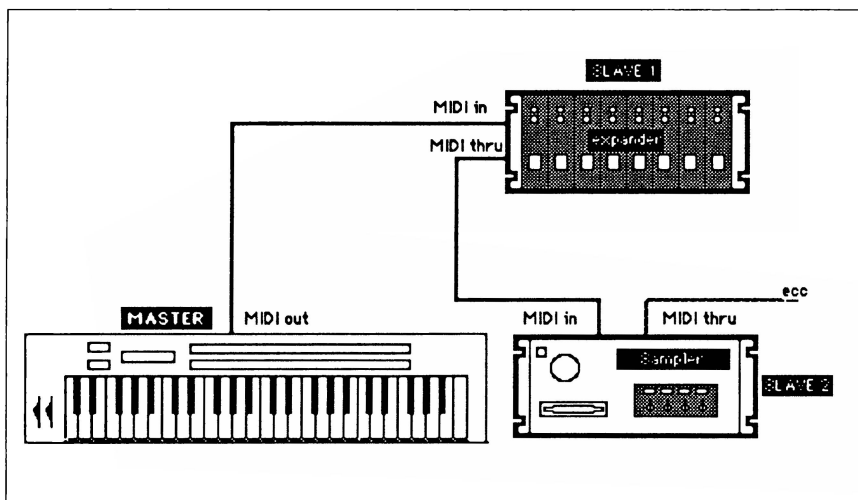



Fig. 2.2 Una architettura MIDI in parallelo.

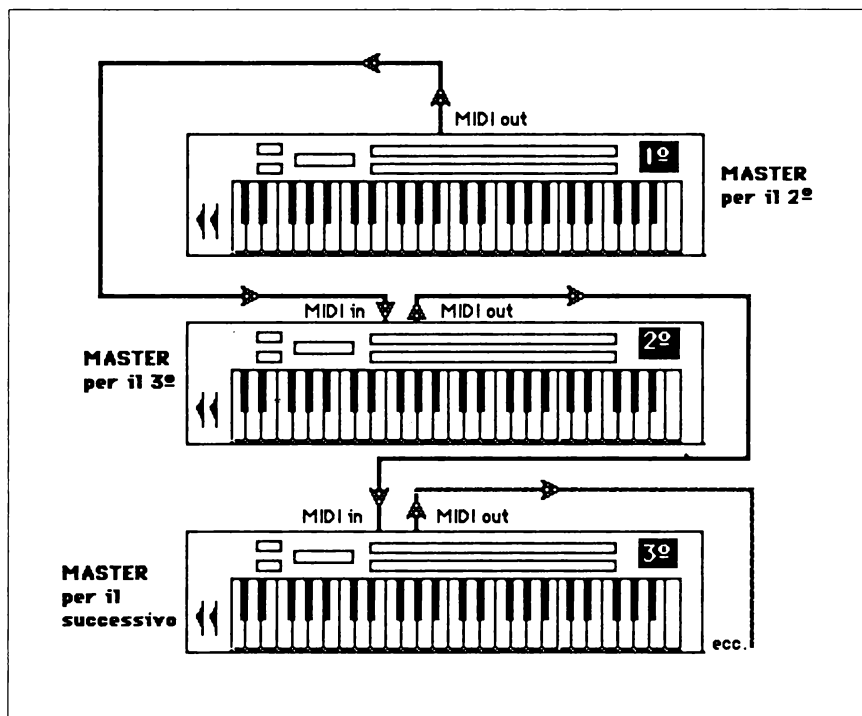



Fig. 2.3 Architettura MIDI in serie.

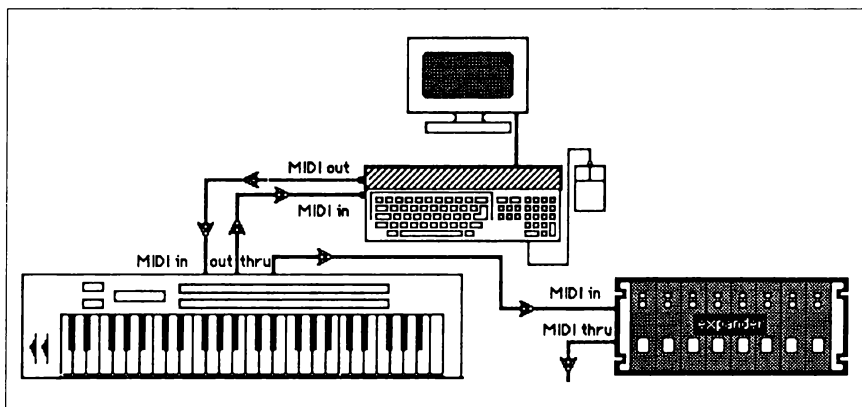



Fig. 2.4 Architettura MIDI ibrida.(parallelo e serie) con Atari ST.

2.5.1. *Il problema dei ritardi*

I ritardi prodotti a causa della bassa velocità della trasmissione e anche per i necessari tempi di elaborazione tanto in ingresso come in uscita dell'informazione, si possono ovviare facendo uso di un moltiplicatore di uscite sullo strumento master (periferica presente nello schema dell'home studio MIDI con il nome di *Thru box*)
Invece i ritardi dovuti alla serializzazione e alla velocità del clock MIDI alcune volte sono limitativi e non hanno soluzione all'interno dell'ambiente MIDI (sono problemi basilari ovvero "di concetto").

2.5.2. *La scelta della scala temperata*

La scelta della scala temperata è un'altra delle limitazioni del MIDI che determina l'obbligo concettuale di uso della scala temperata e quindi il dover ricorrere al pitch bend per avere intervalli microtonali o deviazioni di frequenza. Comunque, in determinate tastiere ed expander, ad esempio EK 44 della Elka e DX7 II della Yamaha, è prevista nella periferica la scelta di altre scale (araba, microtonale, non-temperate ecc.) .
Con la codifica MIDI questo non è previsto in modo diretto però questi fabbricanti hanno assegnato un valore al controllo MIDI diverso da quello temperato.
Con i campionatori si possono configurare scale usando i "programmi", prima dando un'accordatura diversa dalla temperata a ogni campione e dopo assegnando questi campioni ai tasti in modo da costruire una scala.
Nella sincronizzazione attraverso MIDI il numero di battute massimo di una *song* è basso (solo 1024).
Infine la limitazione a 16 canali si rivela sempre più insufficiente. Tuttavia lo standard MIDI è in continua evoluzione e i problemi accennati sono superabili tranne per la banda trasmissione dati, la quale richiede una certa ristrettezza poiché altrimenti perderebbe la compatibilità con gli altri strumenti.

2.6. **Configurazione ideale di un home studio MIDI**

Oggi grazie al MIDI è possibile fare musica di alta qualità tecnologica e sonora con un costo relativamente modesto. E quello che è più rivoluzionario, è che si può fare "a casa".

Il MIDI permette oggi la creazione di un *Sistema per la elaborazione musicale* (SEM) di carattere domestico.
Questo SEM nella letteratura della computer music è noto come "lo studio domestico MIDI" ovvero l'*home studio MIDI*. È questa la linea dove i fabbricanti stanno spingendo di più (per motivi logici di commercialità).
Vediamo adesso una configurazione avanzata di home studio: il cuore del nostro home studio MIDI è l'elaboratore Atari ST.
Ci sono diversi tipi di connessione tra i diversi apparecchi: nel grafico riassuntivo di pagina 97 si vedono differenziate quelle MIDI e quelle non MIDI.
Analizziamo la configurazione di questo home studio seguendo ognuno degli elementi che lo compongono (i numeri in neretto fanno riferimento ai numeri sul disegno).

1, 2, 3 sono gli elementi fondamentali di controllo informatico:

1: l'*elaboratore*

2: il *video display* ovvero lo schermo di comunicazione

3: il *mouse* — che è la "bacchetta del direttore" di questa orchestra informatizzata.

4, 5, 6, 7, 8, 9 sono le periferiche MIDI:

4: la *MIDI thru box* è una periferica inizialmente non indispensabile perché significa solo un risparmio di cavi e di giochi con i *thru* dei singoli apparecchi MIDI.
Però quando ci sono tanti apparecchi MIDI, diventa necessaria per altri motivi: il MIDI è una codifica seriale (come già sappiamo) e tra un messaggio e l'immediato successivo al minimo sono necessari 320 μsec. Quando la quantità d'informazione da trasmettere per essere eseguita "contemporaneamente" è molta — più di dieci messaggi — la sequenzialità diventa un ostacolo perché l'ordine sequenziale diventa apprezzabile per l'udito umano, che ha una risoluzione di 100 millisecondi.
Le *thru box intelligenti*, ovvero quelle più avanzate, hanno un microprocessore di gestione proprio per evitare i "ritardi MIDI" e dunque diventano "necessarie" in una configurazione professionale.

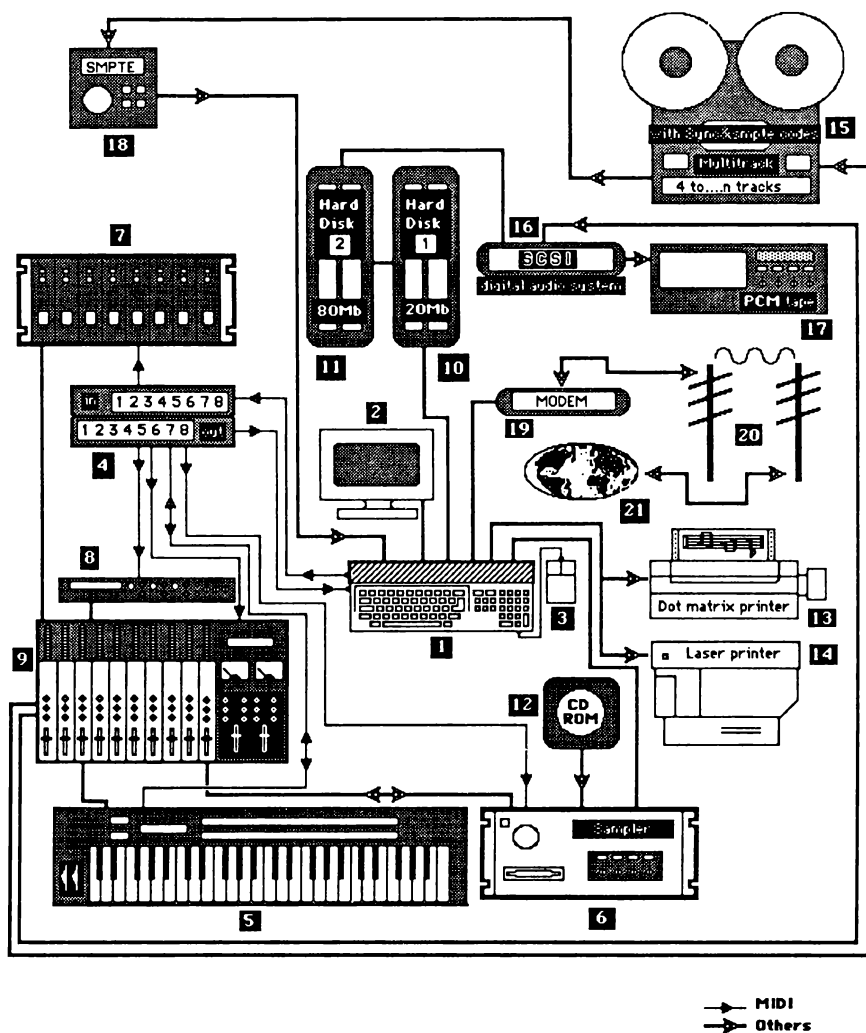



Fig. 2.4 Organizzazione di uno “home studio” MIDI.

5: *La master MIDI-keyboard (tastiera madre-MIDI)*

È una tastiera "totalmente" MIDI — può servire qualsiasi tastiera che lo sia, ad esempio: le tastiere della più parte dei sintetizzatori digitali e dei campionatori MIDI.
Il suo scopo è l'immissione dei dati MIDI; è anche *controller* in tempo reale di tutta la configurazione MIDI dell'home studio — e per questo è l'unico apparecchio, a parte l'elaboratore, che abbia un doppio collegamento (input-output) con la *thru box.*
Oggi la funzione della master MIDI-keyboard può essere assolta anche da una MIDI master guitar, il che può risultare molto più comodo per chi conosce meglio la tecnica chitarristica.

6: *i sampler (campionatori)*

Questi apparecchi sono estremamente interessanti in quanto possono trasformare un suono analogico (ogni suono naturale lo è), immesso attraverso micro o linea, in una serie di numeri (ovvero digitalizzarlo) e possono anche realizzare il processo reciproco, trasformare cioè una serie numerica in un suono analogico.
Un suono elaborato in questo modo riceve il nome di *campione* o *sample.*
Il primo processo di conversione viene detto *A/D*, Analogico a Digitale; il secondo invece: *D/A*, Digitale ad Analogico .
Questa tecnica permette agli elaboratori di "lavorare direttamente con il suono".
Il trasferimento dell'informazione digitale si può verificare attraverso la MIDI e anche senza la MIDI.
È per questo che il campionatore ha un collegamento oltre a quello MIDI con l'elaboratore — quest'altro collegamento si realizza normalmente attraverso la porta seriale RS232.
Il suono digitalizzato non è più che una serie di numeri, dunque è molto semplice modificarlo con l'elaboratore, modificando questi numeri.
Grazie a questi campionatori gli elaboratori possono — con l'aiuto di software specifico — processare e anche "inventare" suoni nuovi, generando serie numeriche che attraverso di loro "diventano suono".

NB: I concetti generali sul modo di lavorare degli elaboratori con il suono vengono spiegati nel capitolo 3 (si veda: *Elaborazione numerica del suono*).

7: *modular synthesizer rack (expander)*

Senza dubbio sono gli apparecchi più popolari, ho voluto mostrarli senza la tastiera, perché concettualmente non hanno niente a che vedere con essa.
Il fatto che i fabbricanti fino a poco tempo fa abbiano proposto i sintetizzatori sempre con la tastiera, non significa che per noi quella presentazione sia interessante.
Quello che ci interessa di più in una tastiera non è quello che ci può interessare di più in un *expander.*
L'abbinamento proposto dai fabbricanti è un fatto che ha a che vedere più con il loro interesse che con il nostro.
Le tastiere MIDI — master di alta qualità — sono costruite con i tasti "pesati" per emulare la *risposta* del pianoforte ed abbinate, in certi casi, a un'altra tastiera *leggera* per la simulazione in MIDI delle tecniche organistiche o di jazz; queste tastiere *master* sono sempre "totalmente" MIDI. Però sono troppo costose per i fabbricanti di apparecchiature "di largo consumo"; dunque, le tastiere dei sintetizzatori di serie sono di qualità "media" (e tante volte non "totalmente" MIDI); lo scopo è ovvio: abbassare il costo dello strumento per renderlo più concorrenziale.
Queste tastiere "economiche" controllano, in un solo *apparecchio*, dei circuiti capaci di "fare suono".
Questi circuiti fanno suono in modo sintetico con delle tecniche precise, e si chiamano tradizionalmente *sintetizzatori.*
I sintetizzatori vengono classificati a seconda delle tecniche con le quali producono il suono.
Le tecniche sono di due grandi tipi: analogiche e digitali (si veda la differenza tra i due concetti nel capitolo dedicato alla codifica MIDI).
Tra i sintetizzatori analogici, ci sono quelli che possono essere controllati con tecnica digitale (attraverso la digitalizzazione dei loro comandi), e quelli che non lo consentono. I sintetizzatori analogici senza capacità di controllo digitale sono rimasti fuori dal mercato, perché non possono essere controllati dal MIDI.
I sintetizzatori controllabili digitalmente che però producono il loro suono con tecniche analogiche costituiscono la famiglia dei sintetizzatori più comuni negli anni tra il 1980 e il 1985 (Moog, Jupiter-Roland, Korg, Yamaha C-XX, Oberheim ecc.).
L'apparizione sul mercato dei sintetizzatori "totalmente" digitali introduce nel mercato il concetto di "modello di sintesi" (per ulteriori chiarimenti vedere nel capitolo 3 *Elaborazione numerica del suono*).

I sintetizzatori digitali per generare il suono lo fanno a partire dall'applicazione di un *modello di sintesi.*
La tecnica di sintesi digitale piu popolare è la *sintesi in FM* (questa tecnica la usano sintetizzatori molto conosciuti come la DX7 della Yamaha o l'EK 44 della Elka ed expander come FB01, FZ 80 ecc.).
Altri modelli di sintesi digitale come la *sintesi additiva o di Fourier* o la *sintesi per funzioni di due variabili* non si trovano implementate in sintetizzatori di largo consumo, dunque sono poco conosciute.
Tornando al discorso precedente la parte del sintetizzatore che produce il suono è "a se stante" ovvero non necessita della tastiera oggi può essere controllata (grazie al MIDI) da qualsiasi altra tastiera-madre — anche da una chitarra master MIDI — o da un elaboratore.
Dunque non è necessario acquistare abbinati tastiera e circuiti, bastano solo i circuiti, e questi vengono oggi presentati in forma *rack* e denominati *expander.*
Normalmente sono modulari e puoi comperarne tanti quanti ne hai bisogno (a un prezzo standard tra i 300 e i 500 US $ l'unità quelli "economici", e tra 1000 e 3000 US $ quelli più costosi).

8: *MIDI controlled digital effects processor* (elaboratore di segnale)

Questo elemento riassume tutto quello che in studio viene chiamato *rack d'effetti.*
Come per tutti gli altri elementi, nel disegno ne abbiamo indicato uno solo per motivi concettuali: puoi usare logicamente più di un *sampler* o più di un rack di *expander*, anche più di un elaboratore, però la configurazione rappresenta gli elementi "concettualmente" significativi di un home studio.
L'*elaboratore digitale di effetti controllato via MIDI* (l'espressione è l'esatta traduzione dall'inglese) è in realtà uno degli elementi più sofisticati a livello tecnologico di tutto l'home studio (di qui il suo prezzo >1000 US $).
Il processore prende il suono analogico del *mixer* e lo processa digitalmente attraverso un DSP (*Digital signal processor*).
Gli algoritmi più comuni di processo sono: *riverbero*, *echo* e *delay*, e anche *trasposizione*, *distorsione*, *filtro*, *loudness*, e fra quelli meno comuni la *spazializzazione.*
Una volta elaborato, il suono viene "restituito" con l'effetto attraverso le uscite analogiche. Il modo di collegamento con il mixer è attraverso le uscite standard *echo send* all'input del processore e dall'output di questo all'ingresso del mixer etichettato *echo return.*

Tutti i parametri e gli algoritmi vengono controllati via MIDI attraverso il *MIDI in* (i primi con il sistema esclusivo e gli algoritmi con il *Programc Change*). Se si lavorasse con il suono digitalizzato tutto questo lo potrebbe fare direttamente l'elaboratore.
Questo però lo spiegheremo più avanti quando vedremo altre configurazioni di SEM (capitolo 3) perché per adesso una configurazione completamente digitale è troppo costosa e non appartiene al capitolo home *studio*.

9: *MIDI controlled mixer*

Tutti sanno che cosa sia un mixer; un mixer controllato via MIDI è un mixer che accetta via MIDI *l'automatizzazione* dei suoi comandi tanto a livello di "volume" come a livello di "tono", come a livello di *switch*.
Il controllo dell'automazione si può realizzare attraverso un programma che controlli la MIDI, ad esempio un programma di *sequencer*.
La "totale" o parziale MIDIzzazione del mixer e logicamente il "protocollo" che accetta come controllo vengono spiegati dai fabbricanti nei manuali d'uso dei singoli prodotti.

Fin qui abbiamo visto le periferiche MIDI.

NB: In un futuro prossimo compariranno nel mercato altre periferiche che useranno la codifica MIDI per rendersi controllabili da un elaboratore o da un master-MIDI; il tipo e le caratteristiche di queste nuove "periferiche MIDI" non sono prevedibili: tutto dipende dell'immaginazione dei disegnatori industriali.

10 , 11 , 12 costituiscono le memorie di massa periferiche dell'home studio:

10: *hard disk da 20 Megabyte*

Un hard disk o *winchester* (disco rigido in italiano) è una memoria di massa periferica che viene collegata all'elaboratore attraverso la porta parallela (il che permette una rapidissima comunicazione, tanto in ingresso quanto in uscita, dei dati)
Questo primo hard disk nella configurazione dello studio è destinato a contenere il software e i file di uso frequente.
La sua capacità è di 20 Mb, ovvero 20 milioni di caratteri, comunque non è molto: lavorando si consumano presto e soprattutto in musica, perché se si

vuole memorizzare suoni (provenienti ad esempio da un campionatore) ovvero *sound files*, questi occupano molto spazio nella memoria.
A 44,1 di frequenza di campionamento 2 Megabyte permettono un campione monoaurale a 16 bit di 22 secondi; se lo vogliamo stereo si dimezza la lungezza del campione e si scende a 11 secondi.

NB: Conviene sempre fare un *back-up* di tutto quello che c'è nell'hard disk, perché gli hard disk sono molto fragili e possono distruggersi facilmente.

11: *hard disk da 80 Mb*

Se lavoriamo con campioni o *sound files* e li vogliamo controllare attraverso l'elaboratore, con programmi *editori di campioni*, e anche se vogliamo generare suoni nuovi via software, applicando modelli di sintesi tipo *FM* o *additiva mediante serie di Fourier*, avremo bisogno di una grande quantità di memoria.
Questo logicamente giustificherebbe il secondo hard disk, ma la sua presenza nello home studio ha un altro significato: questo hard disk deve necessariamente accettare il protocollo SCSI ed è per questo che non solo è collegato al computer come l'altro hard disk, ma è collegato anche a un altro elemento, il n. 16, che è un *digital audio system* con il protocollo SCSI.
Attraverso quest'altro elemento i campioni o *sound files* possono essere direttamente letti, ovvero richiamati dal disco.
Ciò permette la lettura e la scrittura diretta di sound files digitali fino alla capacità massima dell'hard disk (i fabbricanti di hard disk offrono adesso, pilotate da driver con il protocollo SCSI, memorie fino a 768 Megabytes).

12: *CD ROM*

CD significa *Compact Disc* e questo dispositivo è fisicamente uguale ai compact disc per ascoltare musica. ROM significa *Read Only Memory* (memoria di solo lettura). "Voilà la difference", questi CD ROM non sono compact disc anche se fisicamente sono uguali a quelli per ascoltare la musica.
I CD ROM sono memorie di massa di enorme potenza — da 200 a 600 Megabytes (da 200 a 600 milioni di caratteri), però *non* si può scrivere su di loro, sono memorie *a sola lettura*.
I CD-ROM vengono fabbricati in serie (come i dischi) dopo un lavoro di "produzione", e quello che offrono sono "suoni campionati" ovvero *librerie*,

in formato normalmente da 16 bit e frequenza di campionamento pari o superiore a 44,1 Khertz (qualità che viene poi portata al formato di uso del campionatore che li sfrutta).
Un CD-ROM si legge con l'apposito apparecchio lettore come nei compact disc per musica, però il "formato" dell'informazione è diverso, come diversi sono purtroppo gli apparecchi lettori dell'informazione, perché i lettori di compact disc per musica li puoi trovare oggi a partire da 250 dollari, mentre i lettori di CD ROM costano non meno di 1500 dollari.
Ci sono pochi, tra l'altro, CD-ROM di suoni disponibili sul mercato, però lavorare con essi è comodissimo e nel futuro questa tecnologia diventerà uno standard di comunicazione "enciclopedica" (per la sua enorme capacità).
Nella nostra configurazione il lettore di CD-ROM viene collegato al campionatore in un solo senso *read only* (solo lettura) e attraverso la connessione che il campionatore ha con l'elaboratore i campioni si possono portare nelle altre memorie di massa — non direttamente perché non ci sono lettori di CD-ROM con il protocollo SCSI.

13, 14: Le stampanti

13: *Dot matrix printer*, anche *Dot matrix graphics printer*

Ovvero la stampante a matrice di punti, o anche stampante grafica a matrice di punti; questa è la stampante più economica e popolare, però attenzione! esiste una stampante simile nella forma, che non è una DMP, ed era molto popolare prima che si diffondessero le stampanti a matrice: è la stampante a *caratteri*. Questa *non* permette la scrittura grafica.
La DMP permette la scrittura di grafici, perché non funziona con caratteri se non a punti, dunque permette di scrivere notazioni musicali (rigo ecc.) e anche altre forme grafiche, ovviamente anche caratteri alfanumerici.
L'uso di una stampante nell'home studio è ovvio.
Le stampante di serie della Atari è una DMP e gli Atari ST accettano lo standard mondiale Epson 80: dunque possono funzionare con altre stampanti.
La qualità nelle stampanti DMP dipende anche dal software nel quale è scritta l'informazione; un grafico realizzato ad alta risoluzione verrà stampato molto meglio che uno fatto a media o a bassa.
Il livello di qualità della stampante DMP dipende dai punti che è capace di riprodurre per pollice; questo lo potrai sapere consultando le indicazioni d'uso della stampante con la quale lavori.

14: *laser printer*

Se le esigenze, in termini di qualità di stampa, vanno oltre le quattro mura di casa, la soluzione di qualità può essere la stampante laser.
È cara, anche se tutto dipende dall'uso. Ad esempio per le edizioni musicali la DMP non rende il minimo di qualità necessario, dunque... (su questo tipo di problemi consultate il capitolo 3).
L'assenza del *plotter* in questa scelta e ovvia: offre una qualita intermedia tra DMP e la stampante laser, e comunque non risolve i problemi di qualità necessari ed è una tecnologia in disuso.

15, 16, 17 e **18** sono le periferiche di registrazione dello studio.

15: *registratore analogico multitraccia*

È qualsiasi registratore multitraccia in possesso di:

- codice SMPTE (se si vuole lavorare nella sonorizzazione di filmati è imprescindibile; inoltre il SMPTE è il sistema più comune e più usato di *orologio di sincronia con il tempo reale*)
- Sync in o Sync out (che è un sincronismo standard diverso dal SMPTE).

L'elaboratore è collegato attraverso l'interfaccia MIDI-SMPTE o attraverso i Sync (nel caso in cui il SMPTE non sia implementato).

16: *SCSI digital audio system* (registrazione diretta su hard disk)

SCSI è il nome di un altro protocollo molto più specifico di problemi informatici. È un protocollo per la gestione dell'hard disk ossia per la costruzione dei suoi *drivers* (piccoli programmi fatti in linguaggio macchina che permettono agli elaboratori di controllare gli hard disk, inviando, richiamando, cancellando o registrando le informazioni).
SCSI risponde alle iniziali di *Small Computers System Interface* ed è un protocollo che non tutti gli hard disk né tutti i computer accettano; comunque come il MIDI sta diventando standard e sarà sempre più accettato.
Nella nostra configurazione l'elemento 16 è un *digital audio system* che è capace di trasformare il suono che riceve dal mixer (analogico) attraverso un

A/D in un campione numerico usando qualità pari ai compact disc, ossia frequenza di campionamento 44.1 hertz e risoluzione di 16 bit. Con la possibilità, grazie allo SCSI, di registrare direttamente su disco rigido il campione numerico.
Comunque il segnale proveniente dal mixer può passare anche a un registratore digitale standard, tipo PCM.

17: *PCM tape (registratore digitale a nastro)*

PCM sono le iniziali di *Pulse Code Modulation* ed è il sistema con il quale si effettua la conversione di un suono analogico in digitale. Oggi questi registratori possono essere stereo (due tracce) o multitraccia.
Il suono viene accettato *analogico* e trasformato in *digitale*, registrato su nastro (tipo videocassetta o nastro aperto), e quando si richiama viene trasformato un'altra volta in analogico con il procedimento inverso.
Il lavoro con questi registratori è simile a quello con i multitraccia analogici.
L'editing si effettua con sistemi "dedicati", comunque oggi esistono anche software che hanno bisogno di configurazioni hardware un po' particolari, che permettono ai personal computer di realizzare queste missioni di edizione elettronica digitale; nel nostro grafico di pagina 97 questa configurazione non è prevista.
Il suono digitalizzato che si trova nei nastri PCM non è leggibile dal computer, nella situazione attuale di sviluppo del software, dunque il computer non può manipolare questo suono e nel nostro grafico l'elemento 17 non viene collegato al computer in alcun modo.

18: *SMPTE sync to MIDI sync converter*

È un convertitore di sincronizzazioni. I motivi del suo uso sono due:

a) quando si lavora con multitraccia (analogici o digitali) la tecnica di registrazione consiste in passaggi di registrazioni successive. Se si hanno strumenti MIDIzzati, l'unica maniera di essere sicuri che tutti gli eventi coincidano nel modo giusto è usare qualche tipo di sincronizzazione.
Normalmente si raggiunge questo obiettivo registrando in una traccia un segnale di sincronizzazione e facendo divenire gli strumenti MIDI "schiavi" (*slaves*) di detto segnale; questo segnale di sync registrato consiste in una serie

di impulsi distanziati tra loro in modo regolare, il che diventa un riferimento "master" di tempo per gli apparecchi MIDI che, invece di seguire il loro orologio interno seguono questi impulsi.
Dunque qualsiasi variazione piccola o grande della velocità del nastro (e questo succede spesso nei registratori ed è il motivo dell'impossibilità di sincronia "manuale") viene seguita dagli apparecchi MIDI.

b) SMPTE è il codice di sincronizzazione standard dell'industria dell'immagine (cinema e video). Le sue iniziali significano: *Society of motion picture and television engineers*. Attenzione perché non è un codice unico, gli americani usano con quel nome tre tipi di codice: uno per la TV a colori, uno per la TV in bianco e nero, un altro per il cinema. In Europa si usano questi tre più un quarto per il cinema, diverso da quello americano
Questi quattro più un altro chiamato MSSMPTE — che è dedicato alla sincronizzazione con il MIDI —, completano oggi il panorama che la SMPTE-MIDI box deve coprire.
L'ultimo, il MSSMPTE è l'unico che garantisca una totale sincronizzazione tra i sequencer MIDI e i multitraccia o i videotape. Perché non è un segnale di impulsi ma la codifica vera e propria in MIDI degli eventi *Stop*, *Start*, *Continue* e *Song position - pointer*.

19, **20** e **21**: Telecomunicazione. Questi elementi sono la "finestra aperta sul mondo" dello home studio

19: *interfaccia MODEM*

Questa è un altra sigla che indica lo standard di *MOdulation DEModulation*; attraverso il MODEM l'elaboratore può essere collegato alla linea telefonica e "dialogare" con altri elaboratori nel mondo.
Come lavora è semplice: il MODEM converte i dati dell'elaboratore in suoni trasmissibili su una linea telefonica, mentre attraverso la linea riceve la risposta "sonora". Decodifica i suoni che riceve e li converte in *dati* comprensibili per l'elaboratore.
Il collegamento si effettua attraverso l'"interfaccia" connessa alla porta seriale RS232, detta anche *porta MODEM*; e un qualsiasi software di telecomunicazione. La velocità di trasmissione dei dati non dipende dall'elaboratore ma, purtroppo, dalla qualità delle linee telefoniche, che in Europa è bassa, nonostante i costi elevati d'uso.

Le interfacce possono essere l'altro ostacolo per la velocità di trasmisione — ci sono interfacce che permettono solo il collegamento a 300 baud (bit al secondo). In un futuro prossimo con la digitalizzazione degli impianti telefonici la comunicazione potrà diventare "ad alta velocità" e per tanto molto meno onerosa.

20: *L'impianto telefonico o di telecomunicazione*

Non solo la linea telefonica è il supporto della comunicazione — senza dubbio è la più popolare — ma anche la TV (oggi si è diffusa enormemente la consultazione dei servizi informatici delle TV), e la radio, servono per collegarsi con il mondo.
Esistono anche reti di telecomunicazione specializzate nella trasmissione dell'informazione digitalizzata. In Europa le reti di telecomunicazione sono però monopoli statali, il che crea difficoltà di chiarezza e di flessibilità in questi campi.

21: *networks, data bases, archives*

Per capire meglio la potenza della comunicazione con il mondo userò un esempio: l'esempio corrisponde alla descrizione dei servizi offerti da un *network* per la musica e per i musicisti molto popolare negli USA chiamato PAN (*The performing artists network*).
PAN funziona su un elaboratore di grande capacità tipo "mainframe", più specificamente su un VAX 11/750 con più di un Gigabyte di capacità di immagazzinamento (un milione di Megabytes). Esiste per l'utente, tramite il telefono, la possibilità di collegarsi a servizi di posta elettronica (*electronic mail*) incluso il servizio telex della rete; inoltre, sono disponibili i servizi SIG (*Special interest groups*) e alcune sotto-reti.
Queste ultime si occupano di ogni aspetto dell'industria musicale: sale d'incisione, editoria , organizzazione di tournée e servizi di prenotazione, sviluppo di hardware e software, tecniche di sintesi musicale, MIDI, acquisto di apparecchiature ed annunci selezionati per ogni tipo di computer o equipaggiamento audio.
Alcuni di questi servizi dispongono di elenchi dettagliati che permettono, ad esempio, di ottenere una lista di tutti gli avvenimenti musicali di una regione per un periodo determinato.

Molti autori sfruttano queste reti per spedire "elettronicamente" i loro articoli alle organizzazioni editoriali.
Chiunque può associarsi a queste reti attraverso un collegamento *on-line*, cioè componendo il numero telefonico della rete ed entrando in comunicazione con il VAX. Una volta ottenuto il collegamento il sistema centrale richiede il codice di accesso (*password*) per permettere i successivi addebiti (i quali normalmente, negli USA, sono effettuati su carta di credito o su conto bancario). Le tariffe usuali per ottenere una "password" sono tra i 50 e i 200 dollari, mentre gli addebiti per collegamenti in linea si aggirano su i 24 dollari all'ora durante l'orario d'ufficio e sui 6-12 dollari nelle ore serali.
Queste reti permettono agli utenti di partecipare a discussioni circolari, oppure di accedere ai "data base" che contengono librerie di suoni per sintetizzatori, suoni campionati o programmi di utilità per il controllo di ogni dispositivo di generazione di suoni esistente.
Per esempio se uno si trova all'altro capo del paese e ha dimenticato a casa il dischetto con i suoi suoni può collegarsi alla rete PAN e recuperare quanto gli serve.
Downloading significa trasferire informazioni via MODEM dal mainframe al proprio elaboratore. *Uploading* significa il contrario.
Su PAN è disponibile un *forum* per sviluppi di sintetizzatori e apparecchi MIDI su cui ognuno può scaricare messaggi di qualsiasi lunghezza e partecipare così a un dibattito sui futuri sviluppi delle applicazioni musicali con i programmatori e i produttori industriali.
Tramite questo forum è possibile ottenere informazioni molti mesi prima della loro pubblicazione.
Il servizio permette l'organizzazione di conferenze mondiali in *real time* i cui partecipanti possono scambiare informazioni senza ritardi.
La disponibilità di reti a livello mondiale è in continua crescita.
Per questo motivo, molti utenti si servono di procedure "macro", ovvero meccanismi di consultazione automatici (per esempio la registrazione in un videoregistratore temporizzata) per scaricare sul proprio elaboratore le informazioni durante la notte, cioè quando le tariffe di collegamento sono molto meno costose.
Le *macro* sono brevi programmi che rendono automatica la sequenza di azione per collegarsi alla rete; esistono programmi che memorizzano la sequenza di tasti premuti per una successiva ripetizione.
È possibile creare una macro che automatizzi l'intero processo di collegamento telefonico, individuazione del file desiderato, salvataggio di informazioni su dischetto e anche invio di informazioni alla posta elettronica così

come aggiornamento delle librerie di suoni o programmi di pubblico dominio. Inoltre, dal momento che i migliori software di comunicazione "ricordano" la sequenza di azioni del collegamento precedente, diventa molto semplice la creazione di routine "macro".
Questa PAN è una tipica *network* (ovvero uno strumento per lavorare collegati in rete).

Oltre ai network, con la possibilità di input-output di informazioni esistono anche archivi o basi di dati ai quali ci si può collegare solo per ricevere informazioni, normalmente di carattere consultativo — ad esempio in Italia il Videotel, le pagine gialle elettroniche ecc.
Hot Lines: sono "segretarie telefoniche intelligenti"; normalmente coprono servizi post-vendita, o di informazione generale su prodotti, attraverso il MODEM, 24 ore su 24.

L'*home MIDI studio* che abbiamo descritto è sostanzialmente una rete locale o LAN (*Local area network*) che utilizza differenti protocolli per la comunicazione tra i suoi elementi: MIDI SMPTE SCSI ASCII e, se fai uso della stampante laser, il PostScript.
L'integrazione di tutti questi sistemi di comunicazione è destinata ad avere forti sviluppi nel prossimo futuro.

2.7. I sistemi di elaborazione musicale SEM

Gli elaboratori forniti di programmi e apposite periferiche, organizzati in modo razionale a scopi musicali, costituiscono un ambiente di lavoro definito sotto il concetto di SEM ovvero *Sistema di elaborazione musicale*.
Le possibilità aperte con gli elaboratori modificano sotanzialmente il rapporto tra il musicista (creatore e interprete simultaneo in questo nuovo "ambiente") e il modo tradizionale di creare (compositore - partitura - esecutore). Nel capitolo 3 analizzeremo dettagliatamente tutti i tipi esistenti di SEM e la funzione degli Atari ST in queste possibili configurazioni.

Adesso continuiamo a vedere questo tipo particolare di SEM organizzato in rete per lavorare in ambiente MIDI per la produzione di musica (fortemente orientata al pop).
Per poter vedere quanto possa essere flessibile la funzione degli Atari ST in questo ambiente, chiariremo il concetto di *workstation* o stazione di lavoro.

2.8. La workstation musicale

Il cuore di una rete locale è l'elaboratore, e quando un elaboratore viene collegato in *rete* diventa una *stazione di lavoro* ovvero una *workstation* di detta rete. Nel nostro caso la rete ha scopo di produzione musicale e dunque la workstation che controlla detta rete si chiama *musicale*.
Tutti gli elementi visti nello schema precedente erano "periferiche hardware" che possono essere controllate dalla nostra workstation musicale Atari ST.

Di che software ha bisogno il nostro elaboratore (workstation) per controllare questa rete?
Prima di tutto è necessario capire anche a livello basilare il modo in cui operano gli elaboratori per evitare ulteriori confusioni.

Processi analogici e processi digitali
Gli elaboratori possono lavorare *solo* con segnali numerici digitali (*discreti*), dunque tutto il segnale analogico (*continuo*) deve essere digitalizzato altrimenti gli elaboratori non sono in grado di elaborarlo..
Il modo in cui le informazioni vengono processate da un elaboratore è attraverso la loro digitalizzazione; ad esempio le lettere dell'alfabeto hanno una corrispondenza numerica codificata sotto la sigla ASCII (grazie a questo i computer possono elaborare le parole).

Processi Input-Output:
L'elaboratore può ricevere dall'esterno le informazioni, processarle e, una volta modificate (a seconda dei nostri scopi), rispedirle: questo procceso è un tipico processo I/O (input-output, ovveroingresso-uscita).

Dati e programmi
Le informazioni si chiamano*dati* e i processi di trasformazione si chiamano *programmi*.
La costruzione di programmi si realizza attraverso *i linguaggi di programmazione*, che sono sistemi grammaticali di organizzazione degli "algoritmi di processo" degli elaboratori.
I programmi per il controllo delle diverse periferiche e dei dati necessari alla nostra workstation musicale costituiscono il *software*.

Elaborazione numerica del suono
Il suono è un fenomeno di natura ondulatoria che può essere rappresentato

come l'evoluzione nel *tempo* di una grandezza (*ampiezza*); la sua evoluzione fisica è di tipo *continuo* sia rispetto al tempo sia rispetto all'ampiezza (dunque è un fenomeno analogico).
L'elaboratore è capace di trattare solo informazioni di tipo *discreto* ed è perciò necessario trasformare il suono in informazioni discrete e viceversa, per poter mettere in comunicazione il musicista e l'elaboratore mediante suoni.
La realizzazione di questa trasformazione consente poi all'elaboratore di agire sul suono per applicazioni di analisi, elaborazione e sintesi. (Approfondiremo questo tema nel capitolo 3.)
Analizziamo adesso in modo concettuale i processi che si producono tra tutti gli elementi della rete nel nostro home studio MIDI. Questo ci aiuterà a capire meglio quanti e quali di essi possono controllare l'elaboratore e in che modo lo fanno.
Voglio ricordare che questa rete (*SEM* o sistema per l'elaborazione musicale) è orientata alla produzione (sfruttando la MIDI) di musica pop in uno studio domestico standard. Nel prossimo capitolo vedremo altre architetture di SEM più flessibili e adatte a scopi più generali.

Nella nostra rete si producono processi di carattere analogico e di carattere digitale. Nei due tipi di processi l'elaboratore si comporta in due modi differenti:

- nei processi di natura "analogica" l'elaboratore può controllare solo le periferiche che producono processi, nel caso in cui queste periferiche accettino di essere controllate dall'esterno in modo digitale, ad esempio se sono MIDI;
- nei processi di natura "digitale" l'elaboratore potrà direttamente controllare (analizzare, elaborare, sintetizzare) tutti i processi.

NB.: il *suono* è un *processo analogico* però se procediamo alla sua digitalizzazione (attraverso processi di campionamento ovvero A/D e D/A) il suono si rende disponibile a essere processato dal nostro elaboratore a qualsiasi livello.

L'home midi studio per problemi di sviluppo tecnologico e di prezzo permette l'elaborazione numerica del suono solo in modo parziale (non si può lavorare in modo totalmente digitale bensì in modo ibrido).
L'elaboratore, per realizzare questi processi, ha bisogno di poter usufruire del software (programmi) adeguato, e in certi casi di utilizzare l'hardware delle periferiche.

A livello concettuale le applicazioni realizzate dalla nostra workstation possono classificarsi in tre tipi:

- Analisi elaborazione e sintesi di informazione musicale significativa (testo musicale).
- Controllo di periferiche (MIDI e altre).
- Analisi, elaborazione e sintesi del suono in modo numerico.

2.8.1. *Avvertenza preliminare sul concetto di "testo musicale"*

Nella filosofia MIDI la musica (il testo musicale) è solo una serie di "sequenze di eventi musicali" che possono relazionarsi tra di loro, essere sincronizzate (avvenire con un rapporto temporale fra di loro costante); questa "sincronizzazione" è l'organizzazione del testo musicale.
La "composizione" di un testo musicale in ambiente MIDI non è altro che la sincronizzazione di sequenze. Questa viene realizzata mediante programmi specifici chiamati *sequencer.*
Questi programmi si articolano fondamentalmente in tre fasi:

- registrazione (o scrittura) delle singole sequenze di eventi MIDI in genere;
- dialogo interattivo con l'utente permettendo la manipolazione delle sequenze già in memoria a scopo di realizzare la sincronizzazione (normalmente imitando il processo di registrazione di un multitraccia);
- esecuzione delle sequenze (ovvero invio di queste sequenze memorizzate alle periferiche attraverso l'interfaccia MIDI).

Il sequencer è il cuore di un studio MIDI; in realtà non serve per l'analisi, elaborazione e la sintesi del testo musicale nel senso ampio del termine, ma è solo un programma che permette il controllo complessivo da parte dell'elaboratore mediante l'uso dell'interfaccia MIDI. È meglio che cominciamo a vedere questi programmi per quello che effettivamente sono, cioè programmi per il controllo di periferiche MIDI: questa è la loro grandezza e anche la loro limitazione.
La MIDI non è una codifica per capire la musica, è una codifica per aiutarci a lavorare con periferiche digitali; musicalmente è troppo limitativa e schematica, la musica non è un gioco statico di sequenze.
La musica è un fenomeno pieno di processi concorrenti (simultanei) — ad esempio l'armonia — di processi deterministici o non (casuali e aleatori).

Questi processi possono essere sincronizzati in modo rigoroso e non (orchestrazione o improvvisazione) con simmetrie elementari (giri di accordi, seconde voci semplici) e con rapporti diagonali complessissimi (fuga, contrappunto, serialismo ecc.).
Gli oggetti musicali possono trovarsi a molti livelli di astrazione, non solo a quello delle note, e possono costituire non solo sequenze. Oggetti musicali non riconoscibili per la MIDI sono: un accordo (non un arpeggio veloce), un pattern ritmico complesso, un Leitmotiv, una forma musicale, ecc.
Per fare musica è necessario capire il discorso musicale nella sua complessità strutturale e a diversi livelli di astrazione; questo avviene la maggior parte delle volte in maniera intuitiva, per la mancanza di un insegnamento complessivo non-limitativo (la gente capisce sempre di più di quello che vogliono fare credere i difensori del semplicismo).
Il linguaggio di comunicazione tra i musicisti viene plasmato nel testo musicale, il testo per comunicare dovrà essere comprensibile, in modo non equivoco, da tutti, pertanto è necessaria una codifica; la più conosciuta è stata finora la *notazione*, ma non è la sola, ne esistono altre.
La MIDI sfrutta i concetti notazionali, ma lo fa in modo più restrittivo, causando un problema di impossibile soluzione.
La notazione già agli inizi del nostro secolo ha sofferto di una profonda crisi, dovuta alla mancanza di una formalizzazione linguistico-operativa che permettesse di lavorare ai diversi livelli di astrazione in cui opera il discorso musicale, con sufficiente non-equivocità e allo stesso tempo con semplicità.
La notazione è una codifica della musica occidentale temperata, per cui non serve per altre culture musicali.
In notazione è semplice capire gli elementi di carattere minimale del discorso musicale (note, pause, misura, battuta, durata), esiste una certa codifica dei rapporti armonici (con le cifre romane), diventa però impossibile lavorare con strutture più complesse perché non hanno rappresentazione (ad esempio, la melodia, i pattern ritmici, le strutture standard di piccole dimensioni come le danze o le canzonette).
Si è sempre tentato di formalizzare i livelli più astratti della musica (quello delle forme musicali tradizionali come sonata, concerto, sinfonia, ecc.) con l'aiuto dell'assegnazione di lettere dell'alfabeto e con una grande quantità di commenti in linguaggio naturale (con tutte le ambiguità che questo comporta).

a) Analisi, elaborazione e sintesi di informazione musicale significativa (testo musicale)

Oggi l'uso dell'elaboratore come "strumento musicale" ripropone il problema

della codifica dell'informazione musicale a tutti i livelli possibili.
L'elaboratore è uno strumento che opera a livello *logico* prima ancora che *fisico*; in altre parole, non si occupa solamente della produzione del suono, ma anche (e soprattutto) dell'elaborazione del testo musicale.
L'elaboratore può effettuare tre tipi di processi:

- *analisi*: analisi musicologica, deduzione automatica di strutture musicali, riconoscimento di proprietà musicali, descrizione compattata della partitura, confronto tra diversi stili, autori o composizioni;
- *elaborazione*: trasformazione di una partitura in un'altra , caricamento, aggiornamento e cancellazione di partiture o di porzioni di partiture, duplicazione, estrazione delle parti, stampa;
- *sintesi*: generazione automatica, semiautomatica o controllata di partiture, a partire da forme di descrizione astratte come modelli formali probabilistici, reti di Petri, grammatiche generative, algoritmi per la generazione di sequenze.

L'analisi, l'elaborazione e la sintesi del testo musicale si effettuano con appositi programmi costruiti agli scopi sopraelencati.
L'archiviazione dell'enorme mole di informazioni, che costituiscono la cultura musicale, richiede l'organizzazioni di basi di dati (*data-base*) che permettano il rapido accesso alle informazioni, con tecniche sempre più evolute.
In un futuro prossimo questo tipo di applicazioni sfrutterà tecniche e sistemi di Intelligenza Artificiale (il data-base di un sistema esperto costituisce la sua base di conoscenza ed è quello che gli permette "imparare").

b) Controllo di periferiche
Il nostro elaboratore può controllare, attraverso le porte che ha con l'esterno (RS 232 o modem, interfaccia MIDI, porta parallela, hard disk, ecc.), il funzionamento di periferiche; effettuando processi di I/O, può controllare la stampante, può inviare o ricevere informazioni via telefonica (MODEM), può inviare informazione grafiche attraverso il video, e può infine controllare le periferiche MIDI attraverso l'apposita interfaccia.
Questi processi vengono svolti dall'elaboratore con appositi programmi che a seconda della periferica ricevono un nome:

1. Telecom (programmi per la telecomunicazione) quelli che controllano il MODEM.
2. Programmi per la stampa, che permettono la stampa delle informazioni,

negli Atari ST c'è un programma residente nel sistema operativo che verifica la stampa del contenuto dello schermo sulla stampante ad aghi premendo i comandi Shift ed Help contemporaneamente (stampanti a matrice di punti Epson 80 o compatibili), però quando si usano periferiche più complesse — stampanti laser, plotter ecc. — i programmi che permettono il loro uso sono di carattere più specifico e non si trovano nella configurazione di base del sistema operativo degli ST.

3. Programmi per il controllo e/o la comunicazione con le periferiche MIDI Sono di diversi tipi a seconda del tipo di funzione che si vuole controllare e del tipo di periferica, fondamentalmente si possono distinguere cinque tipi:

   - Sequencer
   - Editori di suoni campionati e Sampel-makers
   - Editori di parametri di periferiche
   - Sincronizzatori-interfacciatori col codice SMPTE
   - Utility tools

4. Altri. (I programmi che controllano determinate periferiche si chiamano *dedicati* e questi sono tanti quante le periferiche che si possono collegare).

c) Analisi, elaborazione e sintesi del suono in modo numerico
Nel trattamento del suono, come per il testo musicale, l'elaboratore svolge tre tipi di funzioni:

- *analisi:* acquisizione (e registrazione) di suoni in forma numerica, riconoscimento di timbriche, analisi del contenuto armonico-spettrale, identificazione di parametri per la sintesi numerica di suoni naturali;
- *elaborazione:* editing, riverberazioni, echi e ritardi artificiali, filtri digitali, distorsioni lineari e non lineari, compressioni, decompressioni;
- *sintesi:* produzione di suoni sintetici innovativi, simulazione di suoni naturali, riproduzione di suoni acquisiti (e registrati) in forma numerica.

Riassumendo: l'analisi, elaborazione e sintesi del testo musicale verrà eseguita attraverso l'elaboratore, accettando l'immissione dei dati attraverso la tastiera alfanumerica, la tastiera musicale o dagli archivi in qualsiasi supporto, dischetti o hard disk e anche con il collegamento MODEM per telecomunicazioni, rendendo i risultati disponibili ai programmi di controllo delle periferiche (nel caso in cui si vogliano sfruttare prestazioni MIDI, stampa del testo

musicale, ecc.) o per archiviare (operazione che si può condurre in modo banale o generando un data-base).
Il controllo delle periferiche verra realizzato dall'elaboratore attraverso appositi programmi per il controllo e/o dialogo (stampa, gestione MIDI, gestione hard disk ecc.).
L'analisi-elaborazione del suono verrà eseguita dall'elaboratore dopo l'opportuna digitazione attraverso il campionatore o il digital audio system SCSI, con programmi per l'edizione del suono e/o la post-produzione; la sintesi (invenzione) del suono averrà con appositi programmi che sfruttano il collegamento con periferiche come il campionatore per produrre un suono, sfruttando "modelli di sintesi".

Questa classificazione concettuale permette di capire meglio le funzioni che svolgono tutte le periferiche nella nostra rete.

2.8.2. *Elaborazione mumerica del suono ed elaborazione del testo musicale con la workstation Atari ST*

Passeggiamo una seconda volta per lo schema dell'home studio MIDI con un percorso di andata e ritorno dal suono al testo musicale e viceversa, e lo faremo vedendo le periferiche standard e i programmi commerciali più conosciuti.
Però questa volta lo faremo alla luce della classificazione appena esposta per capire ancora meglio cosa fanno e cosa faranno prossimamente i nostri ST.
Supponiamo di voler cominciare con l'"acquisizione digitale del suono".
Nella nostra cofigurazione questo è possibile attraverso diverse periferiche, ad esempio:

1. La periferica 6 usa la tecnica di campionamento ovvero "sound sampling".
Mediante il campionamento, un qualsiasi suono diventa "dati" e dunque "processabile da un elaboratore ", il processo è semplice: con un microfono o con la linea "campioniamo" con la periferica 6 un suono qualsiasi (ovvero procediamo alla conversione A/D); il campione residente nella periferica verrà trasmesso alla nostra workstation per essere elaborato.
Esistono per gli Atari ST diverse periferiche "esclusive" di acquisizione: una piccola scheda di acquisizione a 8 bit collegabile alla porta MODEM che svolge questa funzione, a un livello di qualità basso e una scheda in rack di qualità media-alta che lavora a 16 bit e frequenza di campionamento variabile fino a 36 kHz chiamata Adap.

2. Un'altra periferica, la 17, immette suono digitalizzato nella rete con una qualità pari ai compact disc: 16 bit lineari e 44,1 kHz di frequenza; è poco conosciuta a livello popolare (tra l'altro perchè è troppo costosa e c'è poco software) questa opera con la tecnica chiamata *disk-based digital recording.*
È la terza volta che in questo libro compare questo metodo di lavoro, la prima volta nella configurazione dell'home studio quando abbiamo parlato dell'uso del secondo hard disk da 80 Mb e immediatamente dopo quando abbiamo spiegato il digital sound system SCSI.
Il sistema di registrazione digitale su disco rigido SCSI è un sistema che si sta popolarizzando poiché permette la qualità dei compact disc o "master digitale" per un tempo limitato solo dalla memoria dell'hard disk.
Ci sono diversi sistemi dedicati già commercializzati, ad esempio l'hard disk della PPG modello HDU e altri.
In un futuro prossimo gli elaboratori che accettano il protocollo SCSI con l'adeguato software e "l'aggiunta di un po' di hardware" (i convertitori A/D e D/A, filtri e in qualche caso anche un DSP, elaboratore digitale di segnali) potranno svolgere direttamente la funzione di registratore "master" su hard disk.
3. C'é ancora un'altra periferica (la periferica 18: registratore digitale PCM) che opera con il suono digitale, però lo fa in modo "chiuso" ossia non lo rende disponibile all'elaboratore.

L'acquisizione del suono comporta subito un secondo processo: *l'elaborazione del suono via software.* Gli Atari ST possono realizzare tre tipi di processi:

- editing di sound file provenienti dall'esterno (via campionatori o schede AD-DA)
- post production sound effects (generazione digitale di effetti come riverberazione, eco, delay, ecc.)
- creazione di sound file applicando modelli di sintesi.

I due primi processi sono processi di *elaborazione di dati.* Il primo, editing di sound files o "edizione di campioni" permette di lavorare con i singoli suoni (campioni) e verificare processi di edizione (*cut & paste*) e di *mix & merge* (addizione proporzionale o sequenziale di suoni).
Il secondo processo è più conosciuto con il nome di "*post-production digital effect*" (processo che si verifica alla fine della produzione nel tradizionale modo di lavorare negli studi di registrazione).

I processi vengono elaborati dall'elaboratore fornito dell'apposito software, la trasmissione dell'informazione avviene con la periferica hardware in modo già spiegato (tanto nella codifica MIDI come nell'uso dei campionatori).
I programmi che permettono il primo processo normalmente tengono conto delle caratteristiche particolari delle singole periferiche, pertanto esistono editori di campioni per ogni tipo di campionatore presente sul mercato, ad esempio per gli Akai S 612, S 900, i Mirage della Ensoniq, gli Emulator, gli FZ1 della Casio ecc.
Il terzo processo invece non richiede l'immissione di dati, *può inventare suoni totalmente nuovi, fare la sintesi del suono via software.*
Oggi cominiciano a commercializzarsi prodotti software per Atari ST che, sfruttando il collegamento con qualsiasi campionatore, "creano" un campione eseguendo un modello di sintesi e trasferendolo poi al campionatore rendendo così disponibile il suo uso immediato; questi programmi vengono chiamati "di generazione del suono per modello di sintesi " o *sampler makers.*

2.8.3. *Un sogno futuribile: il mixing digitale*

Il "mixing digitale" è un'approssimazione alla metodologia di lavoro concepita fino adesso in studio di registrazione analogico; bisogna dunque fornire una piccola spiegazione.
Immaginiamo che tutto quello che abbiamo sono sequenze di numeri; la somma di due suoni corrisponde a una nuova serie di numeri costruita in base alla somma dei valori dei singoli suoni; questa nuova somma può essere diretta o proporzionale a seconda del nostro scopo (due suoni uguali a quelli che sono stati prodotti oppure in proporzioni diverse, secondo necessità) e questo in modo dinamico, ossia variando la proporzionalità liberamente da un istante a un altro.
L'operazione di mix digitale diviene una funzione di calcolo numerico. Tutti gli effetti divengono operazioni su una serie numerica, e queste operazioni potranno esprimersi con un algoritmo o più semplicemente in qualche caso con una funzione.
Dunque il mix digitale diviene un'operazione software che potrà essere complessa quanto si vuole.
I potenziometri se vogliamo possono diventare controlli di "operatori".
Partendo dell'enorme quantità di numeri che corrispondono ai sound-files, la potenza di calcolo dovrà logicamente essere alta e le velocita di calcolo, se non vogliamo aspettare una vita per un mix, dovrà essere altissima , a meno che non

operiamo con diversi processori simultanei, ma questo implicherebbe la complicazione dell'architettura hard-soft dell'apparecchio.
Logicamente un mix in tempo differito può diventare accessibile a un livello basso di spesa. Un mixer digitale deve necessariamente operare in *tempo reale* (*real time*), i tempi di calcolo non devono essere percepibili all'uomo; siamo di fronte a macchine neccessariamente (almeno in questo momento) costosissime.
Fatta questa considerazione, il mixing digitale nella nostra rete non si può fare, rimane come una possibilità concettuale, lontana almeno dagli elaboratori personali se si vuole operare in tempo reale. Esistono tuttavia architetture che prevedono la workstation come terminale in simulazione di un vero studio totalmente digitale.

2.8.4. *I processi di edizione parametrica di periferiche MIDI. Patch editing ovvero edizione di parametri*

Come abbiamo visto quando parlavamo dei sintetizzatori (parzialmente o totalmente digitali) i parametri di costruzione del suono si possono controllare attraverso la MIDI con un elaboratore.
I programmi di editing di parametri, che già sono stati indicati come uno degli standard di programmi per il controllo di periferiche MIDI, non si devono intendere come programmi di elaborazione del suono a livello numerico, perché generano un file che non è un sound file ma è la sequenza di parametri necessari per la creazione del suono in un determinato sintetizzatore.
Normalmente questo è un risparmio enorme di memoria (un "patch file" occupa pochissima memoria — da 500 bytes a 1 K — rispetto all'enorme lunghezza dei "sound files" che occupano da centinaia a migliaia di Kb).
Logicamente si può anche memorizzare la configurazione dei banchi dei sintetizzatori e si possono anche riconfigurare in pocchisimo tempo secondo il desiderio degli utenti.
La trasmisione si verifica attraverso la MIDI nei due sensi: I/O.
I prodotti software editori di parametri sono molto legati al modello di perferica che devono controllare.
Dunque esistono per la maggior parte di periferiche programmi specifici di *patch editing*. Questi programmi vengono normalmente abbinati alle loro "librerie".

2.8.5. *Archiviazione dei dati, librerie e problemi conessi: supporti di immagazzinamento e software di data-retrieval (recupero dei dati)*

Le informazioni hanno diversi problemi quando sono in grossa quantità: necessitano assolutamente di un'organizzazione sicura rispetto ai diversi modi di registrazione (possono provenire da archivio o da librerie). Possiamo distinguere tre tipi di librerie:

- Librerie generate dagli utenti
- Librerie standard
- Attenzione si avvicina una valanga di suoni! Librerie di suoni fatte dagli utenti.

Le librerie di suoni sono dati organizzati in file prodotti dal lavoro di musicisti mediante i processi precedentemente spiegati (campionamento o generazione ed editing o *patch creation*), l'esistenza di una grande quantità di suoni ne impone una classificazione. Quanti più suoni ha un determinato strumento, tanto più atraente diventa il suo uso; tuttavia la sua gestione obbliga a una disciplina di organizzazione. Le librerie e i software per gestirle sono un eccellente strumento di lavoro perché permettono più agevolmente la classificazione.
Le librerie sono conosciute anche come "biblioteche di suoni".
Esiste un gioco commerciale con le librerie di suoni campionati e patches, la gente ha bisogno di loro per lavorare e cerca di procacciarsele in tutti modi.
Ho sentito di dischetti di suoni venduti a prezzi superiori a quelli del software: mi pare un po' esagerato ma è anche normale.
Probabilmente la creazione di librerie da parte di una persona competente comporta un lavoro e una sensibilità che meritano di essere pagati, soprattutto perché si risparmia tempo e si crea nuovo lavoro.
Non esiste alcun mezzo legale per evitare la copia di questi prodotti, perché per tradizione vengono considerati "di dominio pubblico".
Però sono troppi quelli che, sfruttando il fatto di accedere alle librerie già di dominio pubblico, cercano di farne un commercio imponendo dei prezzi ingiustificati.
Il prezzo pagabile per le librerie è quello giusto "per un servizio" ma non dell'ordine di grandezza del prezzo di un software.
Vorrei riflettere un momento sul supporto abituale delle librerie e dei suoni: il dischetto da 3.5 pollici, che è diventato lo standard.

2.8.6. *Dati, supporto, problemi di trasmissione*

L'informazione, ovvero "i dati" risiede e viaggia in un supporto, che può essere di diversi materiali:

- Magnetico: dischetti, dischi rigidi (chiamati anche hard disk o Winchester), cassette, streaming tape (specie di casseta più grande), nastro ecc.
- Ottico a lettura laser: CD-ROM, WORM...
- Silicio: firmware, ROM ecc.
- Carta: Softstrip, altri...

Ognuno dei supporti ha i suoi vantaggi e i suoi inconvenienti però senza dubbio il supporto standard oggi per l'informatica musicale a livello popolare è il dischetto magnetico doppia faccia doppia densità da 3,5 pollici.

I dischetti da 3,5 pollici possono archiviare suoni o patches *in diversi formati,* il che, quando si lavora con apparecchi diversi che accettano driver da 3,5 pollici, genera delle confusioni enormi.
I dischetti di una periferica sono solo di questa periferica; infatti se hai una libreria per DX7 in formato DX7 (ovvero in dischetto leggibile dal suo "driver") questo dischetto non può essere letto dal tuo Atari ST, e viceversa.
Il modo di trasferire le librerie dall'elaboratore alla memoria interna della periferica è via MIDI: mai mettere il dischetto che contiene la libreria in formato Atari nel driver della periferica o viceversa.
Vediamo lo schema di comunicazione dei dati delle librerie nel nostro esempio precedente. In questo esempio vediamo il processo per trasferire i dati che ci sono nel dischetto A al dischetto B.
I dati viaggeranno attraverso la MIDI. L'elaboratore controlla il processo con un software Patch/Librarian, i dati verranno richiamati via software alla periferica mediante dei messaggi MIDI che usciranno dell'elaboratore attraverso la MIDI out, e la periferica li riceverà attraverso la sua MIDI; una volta elaborato il messaggio, la periferica spedirà all'elaboratore, attraverso la MIDI out, i dati desiderati e l'elaboratore li riceverà dal suo ingresso MIDI, cioè attraverso l'interfaccia MIDI; una volta decodificati, i dati entreranno nella RAM dell'elaboratore e infine saranno salvati sul dischetto B.
I campioni possono viaggiare attraverso la MIDI, attraverso le porte seriali e le porte parallele.
Quando si trovano nel campionatore (sfruttando il suo formato) si chiamano campioni, quando si trovano nell'elaboratore (ovvero nel suo formato) si

chiamano “sound files”.
I dati che operano nel campionatore sono *campioni* o *sample*. Quando vengono trasmessi all’elaboratore sono riformattati, e il risultato di questa manipolazione in quel momento è un “sound file”.
Quando si rispediscono alla periferica per essere elaborati ritornano al formato della periferica e dunque ritornano a essere campioni o sample.
Un dischetto di sound files non può essere letto da un campionatore e un dischetto di campioni non può essere letto da un elaboratore.

Il trasferimento di grandi quantità di informazioni mette in primo piano i problemi di *velocità di trasferimento*. Facciamo un esempio che vi mostrerà dove si trovano i problemi e le diverse soluzioni oggi.
Già abbiamo spiegato il concetto di “baud rate” adesso cerchiamo di applicare il concetto alla pratica.
Se la velocità di trasferimento di un campione (che ha una grandezza ad esempio di 250 Kb) è di 300 baud (ovvero 300 bit al secondo, velocità di comunicazione dei modem economici), facciamo un po’ di conti:

baud = 1 bit al secondo
8 bit = 1 byte
1 Kbyte = 1024 byte.

250 Kb × 1024 = 256.000 byte

256.000 × 8 = 2 048 000 byte

2.048.000 / velocità di trasmissione (qui 300) = 6.826, 66 secondi

il che, tradotto nelle normali unità di misura del tempo, significa: 1 ora, 53 minuti, 46 secondi e 7 decimi !!!!
Ciò significa che procacciare questo campione via MODEM con un baud rate basso ci costerà una bolletta pari al valore dell’interfaccia(!). La MIDI trasmette a una velocità di 31.25 Kbaud, dunque non tenendo conto dei messaggi MIDI che accompagnano il trasferimento del campione (che occupano comunque un po’ di spazio) e seguendo l’esempio precedente:

2.048.000 / velocità di trasmissione (qui 31.250) = 65, 536 secondi
ovvero: 1 minuto, 5 secondi e 54 centesimi

Questo è il tempo che dovremo aspettare per trasferire il campione da un software di editing via MIDI dall'elaboratore al campionatore e viceversa.
Un minuto è una discreta quantità di tempo, però l'ordine di grandezza di questo campione non è esagerato. Il trasferimento di tutto il contenuto di un dischetto via MIDI richiede almeno tre minuti, istruzioni a parte.
Invece i driver dei floppy disk funzionano con una velocità di trsferimento di 250 Kbit. Continuando con il nostro esempio: questo campione sarà caricato da dischetto in memoria in un tempo pari a:

> 2.048.000 / velocità di trasmissione (qui 250 000) = 8.192 ovvero 8 secondi e 19 centesimi.

Che è quello che devi aspettare per caricare nel tuo computer un "sound file" da un dischetto Atari e anche per caricare un campione da un dischetto del campionatore.
Il contenuto di un dischetto completo sarà caricato in memoria in un tempo uguale a ventitre secondi e sei decimi.
Questa velocità è ancora troppo bassa per trasferire archivi dell'ordine dei CD-ROM, 200 o 600 Mb.
Le interfacce parallele trasmettono a una velocità dell'ordine di d[illegible]ne o centinaia di Megabit al secondo. Specificamente l'interfaccia paralella a cui si collega l'hard disk dell'Atari ST funziona a un baud rate di 10 Megabit. Il "sound file" del nostro esempio sarebbe caricato dall'hard disk all'Atari ST in un tempo pari a:

> 2.048.000 / velocità di trasmissione (in questo esempio 10 000 000) = 0.2048 secondi, ovvero in due decimi di secondo !

Così, in funzione della velocità di trasmissione permessa dal supporto, per uno stesso dato è possibile che tu debba aspettare quasi due ore di telefono o solo due decimi di secondo, a seconda che tu lo voglia caricare da MODEM oppure che debba passare attraverso una porta parallela da un CD-ROM o un hard disk.

2.8.7. *Le librerie di grande capacità di memoria su supporto CD-ROM (librerie di campioni e patches)*

Già sappiamo che per i campioni il CD-ROM è un supporto ideale per la sua grande capacità (da 200 a 600 Mb), analizziamo adesso però l'offerta nel

mercato musicale di librerie su questo supporto. Il lavoro di ricompilare in un CD-ROM campioni e patches è pesante e costoso, ma il livello di qualità dei campioni è normalmente alto, per compensare lo sforzo.
La valutazione dell'interesse (qualità-prezzo) di questi prodotti, come per qualsiasi altra offerta simile sul mercato, è assolutamente personale. L'offerta oggi è molto bassa, esistono pochi apparecchi (Emulator, Kurzweil ecc). *Comunque con l'apparizione dello standard di campionamento a 16 bit l'offerta si moltiplicherà sicuramente.*

2.8.8. *Creazione di una musica eseguibile*

Attraverso i processi che abbiamo elencato, di creazione dei *sound files* o dei *patches*, la rete ha "sotto controllo" la materia prima della musica (i diversi suoni).
Per fare musica dovremo organizzare una "composizione" che sfrutti questi suoni, ovvero dovremo generare un "testo musicale".
Il testo musicale nella mente dei musicisti è sempre la combinazione di diversi oggetti musicali, a diversi livelli di astrazione.
Molte volte qualcuno di questi oggetti musicali generatori di una nuova composizione nasce fortuitamente, "improvvisando", quando si sta suonando uno strumento; altre volte sono solo idee; altre volte sono variazioni su "pattern conosciuti", che è necessario "plasmare", cioè formalizzare.
La nostra rete permette al musicista di lavorare "naturalmente" nei modi appena visti.
A livello dei "dati" iniziali per generare una composizione ci sono due possibilità con gli elaboratori:

- Immissione dei dati (oggetti musicali primitivi)
- Generazione automatica o semi-automatica dei dati.

L'*immissione dei dati* musicali nell'elaboratore può avvenire in:

- *Real time input* (registrazione MIDI di una performance, via tastiera MIDI (drums-MIDI, Micro-Pitch to MIDI converter, o chitarra MIDI).
- *Step by step input* (registrazione MIDI in tempo differito dalle stesse periferiche — un'informazione dietro l'altra assegnando interattivamente da programma qualche parametro — normalmente le durate).
- *Input attraverso notazione o codifiche* (scrittura con un programma per

notazione, o qualsiasi altra codifica; l'immissione dati avviene via tastiera alfanumerica o sfruttando interfaccia grafica e mouse).

- *Pattern recognition* (i dati vengono letti dall'elaboratore attraverso una periferica — telecamera ad alta risoluzione o scanner ecc. — e vengono interpretati da un programma capace di riconoscimento di caratteri).

La *generazione* può essere:

- *Generazione algoritmica* (i dati vengono generati non dall'utente ma direttamente dall'elaboratore; in questo modo l'utente può influire sui dati in modo indiretto ovvero modificando parametri che influiscono sul comportamento dell'algoritmo in modo interattivo).
  L'individuazione di algoritmi a diversi livelli di astrazione per la creazione di musiche a partire da processi ad alto livello è uno dei campi di ricerca più seri in questo momento nella musicologia e nell'informatica musicale mondiale.
  Questi algoritmi permetteranno ai musicisti, in un futuro, di "creare musica" a partire da una idea astratta, senza bisogno di tecniche noiose e ripetitive, solo grazie alla loro intelligenza e sensibilità.
  Il musicista, partendo da un'idea generale di un processo musicale — dovendolo rappresentare con dei formalismi grafici evocativi (al modo di un diagramma di flusso) — potrà generare con l'aiuto dell'elaboratore, in modo altamente automatizzato però decisamente interattivo, da suoni nuovi a nuove musiche, da idee generali a partiture dettagliate scritte in notazione o altri sistemi, da musiche per un solo strumento a orchestrazioni.
  Con l'uso di questi algoritmi ci si potrà permettere il lusso di "provare" sulle idee originali tutti i giochi formali culturalmente conosciuti ed anche giochi aperti, cioè con possibilità di non determinismo.
- *Generazione non-algoritmica* (metodo usato nelle applicazioni d'Intelligenza Artificiale, ove i dati vengono generati automaticamente dall'elaboratore applicando un insieme di regole a una base di conoscenza).

Una volta immessa l'informazione musicale nell'elaboratore questo potrà effettuare i processi sopraelencati: analisi, elaborazione e sintesi del testo musicale.

Analisi

I processi d'analisi a scopo di composizione possono essere tantissimi, qualcuno ha più senso sotto il profilo musicologico (aspetti statistici, segmentazione di melodie), altri invece possono essere di grande utilità per successive elaborazioni (individuazione di pattern ritmici espliciti e nascosti, analisi armonica tradizionale o non tradizionale, verifica degli accenti.

Elaborazione

L'elaborazione degli oggetti musicali primitivi permette la creazione vera e propria del brano.
L'elaboratore può, via software, trasformare qualsiasi dato in un altro in modo parametrico, duplicarlo, frazionarlo ecc.
Di tutte le trasformazioni parametriche possibili solo una quantità abbastanza ridotta ha un senso musicale (ad esempio trasposizioni diatoniche o tonali), il che permette di generare in modo automatico o semi-interattivo armonizzazioni a diverse voci, applicazioni di una melodia (come rapporto intervallare) a un altro pattern ritmico (modifica delle durate originali).
Queste trasformazioni parametriche possono essere sottomesse anche a regole di priorità, dominio (*range*), soglie ecc.
Il risultato di un'elaborazione a un livello di astrazione può essere la base per ulteriori elaborazioni a più livelli; il che permette di realizzare, con l'aiuto dell'elaboratore, orchestrazioni, arrangiamenti in genere ecc..

Sintesi

La sintesi dei lavori di analisi e di elaborazione degli oggetti musicali sarà una composizione che attraverso un programma di "edizione" o, come si dice più spesso in ambiente informatico, di *desktop publishing* realizzerà la "composizione dell'originale" e fornirà alle periferiche di stampa (DMP o laser) il testo musicale pronto per essere stampato in versione integrale e anche con le parti separate a seconda delle necessità.
Tutti questi processi verranno guidati in modo interattivo dall'utente e realizzati da programmi disegnati a tali fini.

2.8.9. *Limitazioni tecnologiche e tempi di elaborazione*

Chiunque abbia sensibilità musicale sa che il rapporto con il suono per un musicista è interattivo. Molte volte l'oggetto musicale che può provocare una composizione è semplicemente la *tessitura* di un determinato suono. Il lavoro in reti come questa permette dunque di lavorare dalla base con il fenomeno musicale, dal suono pre-musicale al discorso musicale.
Purtroppo, a livello tecnologico i miliardi di possibilità di sfumature interattive tra l'uomo e lo strumento ancora non sono raggiungibili dalla computer music in tempo reale.
Un musicista che suoni qualsiasi strumento sfrutta centinaia di anni di cultura su quello strumento (violino, flauto, voce...) e qualcosa in più della tradizione culturale; tutti gli strumenti rispondono in tempo reale al loro interprete; tutti i musicisti hanno più o meno consapevolmente un rapporto "personalizzato" con il loro strumento. Detto in linguaggio analitico: "Ogni strumentista fa suonare il suo strumento in modo personale".
L'elaboratore è capace di "emulare" tutto quello che sia formalizzato ovvero che noi siamo capaci di dirgli di fare. L'ottimizzazione del rapporto musicista-elaboratore per applicazioni in tempo reale, tuttavia, è ancora nella preistoria.
È opportuno a questo punto chiarire bene che cosa siano il *tempo fisico*, il *tempo reale*, il *tempo nel rapporto interattivo* e il *tempo nel rapporto differito*, concetti che finora abbiamo usato un po' implicitamente.

Tempo fisico: è il tempo nel senso scientifico della parola e la sua misura è ben nota: anni, mesi, giorni, ore, minuti, secondi, millisecondi, nanosecondi ecc.

Tempo reale: è un concetto "umano" dipendente delle capacità percettive degli essere umani. Noi vediamo 15 volte al secondo, ascoltiamo 10 volte al secondo. Un esempio molto esplicativo: il cinematografo è una successione discreta (discontinua) di 24 immagini al secondo, però noi vediamo una continuità determinata dal fatto che memorizziamo solo 15 volte al secondo. Anche la TV è discreta e noi la vediamo continua.
Tutti gli elaboratori fino ad ora sono sequenziali (fanno solo una cosa alla volta) però lavorano ad alta velocità. In ambiente informatico esiste una misura chiamata IPS (istruzioni al secondo) che indica la velocità di processo di una singola istruzione per la CPU di un elaboratore: per IPS = 1 il computer elaborerebbe un'istruzione al secondo; per IPS = 10.000 elaborerebbe diecimila istruzioni al secondo .
I computer svolgono il loro lavoro elaborando istruzioni; se un determinato

processo è realizzabile dall'elaboratore sotto la capacità percettiva dell'utente umano, è realizzato *in tempo reale.*
Per dirla in parole povere, se l'elaboratore può (pur essendo sequenziale) mostrarsi, di fronte all'uomo, capace di fare diverse cose alla volta è perché l'uomo sotto una soglia di tempo fisico non percepisce niente. Il tempo fisico minimo della percezione umana marca la frontiera delle applicazioni in tempo reale.
Ovvero: tempo reale = sensazione di inmediatezza.

Tempo che permette il rapporto interattivo: è un concetto anche umano. Quando i tempi di elaborazione sono modesti (secondi o minuti) in una singola sessione di lavoro si può verificare tra l'elaboratore e l'utente un rapporto interattivo. Un esempio tipico di interattività è sviluppare una formula algebrica complessa con una calcolatrice tascabile, dove l'utente formula le singole operazioni che possono essere automatizzate dalla calcolatrice nell'ordine che vuole, riceve i singoli risultati e con questi realizza altre operazioni con la calcolatrice.
Con gli elaboratori il rapporto è sostanzialmente lo stesso, però con la possibilità di memorizzare i risultati nell'ambito dello stesso elaboratore.

Tempo differito: quando i tempi di calcolo sono alti, pari o superiori alla durata di una singola sessione di lavoro, si parla di *elaborazione in tempo differito.* Oggi la velocità di processo e la memoria degli elaboratori sono cresciute molto (e continuano a crescere rapidamente), e questa modalità di elaborazione si rende sempre meno necessaria, però nella storia dell'informatica musicale i processi di elaborazione numerica del suono sono stati finora in questo ambito. Nella nostra rete qualche processo di elaborazione numerica del suono (lavorando con editing di suoni o sampler makers) potrà a volte impiegare anche più di dieci minuti, il che può spaventare l'utente abituato a un rapporto interattivo spinto; d'altronde, le limitazioni tecnologiche sono queste.

2.8.10. *Applicazione alla rete di home studio MIDI*

La nostra rete prevede la realizzazione, con l'aiuto dell'elaboratore, della produzione di un brano ibridamente tra il tempo reale e la modalità interattiva:

a) Interattivamente (e anche in qualche caso in tempi differiti) si elaborano i singoli suoni.

b) Interattivamente si elaborano il testo musicale e la sua stampa.
c) In tempo reale si realizza la performance del testo che viene o registrato nelle singole tracce del magnetofono multitraccia o registrato solo col segnale sync in una singola traccia di sincronia, il che permetterà la realizzazione sincronizzata in tempo reale della parte assegnata all'elaboratore con il resto delle singole tracce del magnetofono.
d) Interattivamente si preparano gli elementi di produzione (mix computerizzato via MIDI)
e) In tempo reale si realizza il master (con l'aiuto delle periferiche di post-produzione, rack effetti controllati via MIDI e sincronizzatori).

2.8.11. *Controllo di periferiche*

La composizione redatta con l'aiuto dell'elaboratore potrà essere eseguita in modo non-automatico con strumenti suonati dal vivo da musicisti, che si serviranno della partitura stampata dalle apposite periferiche della rete.
L'elaboratore può anche automatizzare l'interpretazione, tutta o in parte. In questo caso è più impegnato nella "produzione di musica".
Nella nostra rete, l'elaboratore può interpretare la composizione o suonando da solo (con il chip interno di suono) o attraverso il controllo di periferiche MIDI che "suonano".
La produzione musicale nella nostra rete segue il modo di lavoro dello studio di registrazione multitraccia dove l'elaboratore diventa schiavo (*slave*), a livello di sincronizzazione, del magnetofono e ha il compito di controllare periferiche.
Il controllo delle periferiche richiede un "testo di controllo", ovvero, l'elaboratore dovrà inviare alle periferiche una serie di informazioni che permetteranno a quelle di rispondere in tempo reale.
La grandezza del MIDI sta nell'integrare in una sola codifica controllo e informazioni che permettono di lavorare con il suono; nell'automatizzare l'interpretazione del testo musicale; nel sincronizzarsi con macchine esterne tipo magnetofoni e nel controllo in tempo reale, a livello parametrico, delle periferiche di produzione e di post-produzione.
Il testo di controllo nella nostra rete è il testo MIDI.

2.8.12. *I programmi sequencer*

Una volta generato il testo musicale, i dati dovranno essere trasformati in formato MIDI per diventare eseguibili. Questa operazione viene svolta normalmente dai programmi *sequencer* abbinati a programmi specifici di scrittura. Il *sequencer*, come abbiamo detto prima, è un programma che permette il controllo complessivo, da parte dell'elaboratore, dell'uso dell'interfaccia MIDI.
Prima di tutto vorrei evitare una confusione terminologica: il sequencer è stato, fino a poco tempo fa, commercializzato come una configurazione chiusa hardware e software: è stato l'elaboratore "dedicato" più popolare dell'informatica musicale.
Oggi, attraverso un programma *sequencer* e l'interfaccia MIDI, qualsiasi elaboratore *general purpose* può svolgere la sua funzione; con in più, logicamente, tutto quello che permettono la potenza dell'elaboratore e l'"intelligenza" degli altri programmi.
Dunque, quando parliamo di *sequencer* non parliamo mai dell'apparecchio conosciuto con quel nome, ma specificamente dei "programmi" che permettono di lavorare con le sequenze di eventi MIDI e di usare l'omonima interfaccia (MIDI) su qualsiasi elaboratore.
Come nel caso del testo musicale, il testo di controllo, cioè i dati che sfruttano le potenzialità del sequencer, fondamentalmente può essere immesso in diversi modi:

- *Real time*: che è registrare — memorizzare — da un master MIDI i dati in tempo reale ovvero mentre si sta operando con la periferica (è cosi che si registrano le sequenze musicale da tastiera o i parametri di mixing da un tavolo di mix-MIDI o computerizzato). A livello operativo i dati generati nel master MIDI vengono trasmessi, attraverso l'uscita MIDI-out, all'ingresso MIDI-in dell'elaboratore, il quale con una procedura software li "registra". I dati poi possono essere ulteriormente elaborati.
- *Interattivamente*: ovvero scrivendo nel sequencer dato per dato.
- *Generazione di dati MIDI con procedure algoritmiche*

Gli elaboratori processano o creano dati con algoritmi precisi: la trasformazione o generazione dei dati mediante algoritmi è un processo fondamentale in tutta l'informatica.
Gli algoritmi possono essere di molti tipi: ci sono algoritmi complessi e algoritmi semplici. I progettisti di software dedicano gran parte del loro sforzo

alla ottimizzazione degli algoritmi di processo per migliorare la "qualità" del software.
Fra i modi di generazione con procedure algoritmiche vi sono:

a) Trasformazione in MIDI di informazioni musicali non-MIDI. La trasformazione dei dati si ottiene applicando un "algoritmo di trasformazione"; è così che un testo musicale scritto in qualsiasi codifica (notazionale o non) può diventare una sequenza di eventi MIDI.
b) Generazione automatica o semi-automatica di dati MIDI. Negli USA si stanno diffondendo dei "giocattoli MIDI" che sono generatori automatici di sequenze MIDI, attraverso l'applicazione di algoritmi generativi. L'utente in questi programmi non ha accesso alla creazione dei dati MIDI in modo diretto, ma può modificare i parametri di questi algoritmi. Il risultato (per un po') è abbastanza divertente, però gli algoritmi utilizzati per questi giocattoli tutt'altro che "intelligenti" sono molto limitativi — l'applicazione di algoritmi di generazione ciclici, ossessivi, o anche "pseudocasuali" — nel secolo XX più che una novità sono una noia!
Invece generare dati a partire da algoritmi è concettualmente importantissimo, come già avevamo preannunciato quando parlavamo dello stesso tipo di generazione a livello di testo musicale.

Dopo l'immissione/generazione delle sequenze di eventi MIDI i sequencer permettono l'elaborazione di questi dati logicamente in modo interattivo e inoltre emettono questi dati attraverso il MIDI-out, il che permette all'elaboratore di controllare le periferiche MIDI della rete. In seguito vedremo come; adesso torniamo invece alla produzione di musica con la nostra rete.
Il sequencer nella prima applicazione in tempo reale dovrà sincronizzarsi con il registratore, dunque vediamo come si realizza questo processo e quali periferiche siano coinvolte.

2.8.12. *Sincronizzazione*

La codifica MIDI prevede la sincronizzazione con due posizioni *master* o *schiavo*; l'apparecchio master ha il rapporto con il tempo fisico, il resto si sottomette al suo orologio o *clock*; qualsiasi variazione di tempo nel master sarà eseguita da parte degli strumenti schiavi immediatamente e così la rete è sincronizzata.
Il master dovrà essere logicamente lo strumento meno affidabile a livello di

rapporto con il tempo fisico. Gli apparecchi meno affidabili sono quelli meccanici e nella nostra rete la scelta cade sul magnetofono. Così qualsiasi piccolo malfunzionamento meccanico non rappresenterà un problema per la produzione, perché tutta la rete risponderà "sincronizzata".
Esiste invece un problema di base: prima del MIDI esisteva già una codifica di sincronizzazione tra macchine meccaniche finalizzate alla produzione di musica e immagine e questa codifica è il SMPTE.
La codifica MIDI oggi a livello di sincronia è meglio della SMPTE, però non è lo standard nel settore. Quando si vuole sincronizzare il suono all'immagine in un filmato si deve utilizzare la codifica SMPTE. La maggior parte dei magnetofoni professionali incorpora o almeno ha a disposizione opzionalmente il SMPTE.
Nel caso in cui il nostro scopo sia solo quello di realizzare musica possiamo lasciar perdere il SMPTE e sfruttare il MIDI come codifica di sincronia. Oggi la tendenza è fare convivere tutti e due (ci sono accordi ad alto livello per sviluppare una codifica mista "standard" MIDI-SMPTE).
Il modo in cui vengono sincronizzate tutte e due è attraverso l'uso di una periferica che si chiama *sync-box* ovvero "sincronizzatore" SMPTE to MIDI *and back.* Questa periferica legge la codifica SMPTE e la trasforma in MIDI permettendo alla rete ad esempio di sincronizzarsi con un filmato già fatto ed anche con una registrazione realizzata solo con SMPTE.
Per lavorare con la sincronia MIDI nel caso in cui il magnetofono non sia SMPTE, il processo è il seguente: si riserva una traccia del magnetofono ove vengono registrate in tempo reale le informazioni di sincronia MIDI emesse dall'elaboratore (in questo caso l'elaboratore è il master). Una volta presente questa traccia nel multitraccia l'elaboratore diventerà sempre schiavo del magnetofono e schiave diventeranno, attraverso l'elaboratore, tutte le periferiche.
Una volta garantita la sincronizzazione la produzione del brano è realizzabile per tecnica multitraccia. Questa ha due passi significativi:

- Registrazione delle singole tracce
- Mix

Vediamo questo secondo passo nella nostra rete.

2.8.13. *Mixer controllato via MIDI*

Oggi è molto comune nel lavoro degli studi di registrazione risparmiare tracce dell'apparecchio multitraccia registrando in tracce separate solo tutto quello che non è MIDI e lasciando una traccia per il segnale di sync; così, poi, in tempo reale si effettua il mixaggio finale sul master delle tracce del registratore e di tutte le tracce delle periferiche MIDI (drum machine, campionatori, expander di sintetizzatori ecc.).
Questo ci risparmia tracce di registratore multitraccia (costose) e rumori di fondo e d'incisione (i registratori analogici sono estremamente rumorosi e si mangiano una buona parte della dinamica potenziale dello strumento).
Aumentano invece i problemi di mix: se si possono risparmiare tracce sul registratore, lo stesso non si può fare con il mixer, e per questo tipo di lavori sono indispensabili mixer con sempre più tracce. L'aumento del numero delle tracce in un mixer rende il suo uso manuale più difficile (immagina di controllare manualmente un mixer a 72 tracce!).
Nell'automazione dei mixer non è una novità l'uso di "computer dedicati": questi danno l'automazione tanto dei *faders* (potenziometri di volume) quanto di altri controlli del mixer.
La "computerizzazione del missaggio" è stata il cavallo di battaglia delle ditte costruttrici di mixer professionali dalla fine degli anni settanta fino ad ora.
Il trionfo dello standard MIDI ha permesso un protocollo universale di comunicazione tra apparecchi digitali e come è successo nel caso dei sintetizzatori, il controllo digitale dei circuiti che operano sui parametri (fondamentalmente filtri e amplificatori di segnale) così come le matrici di incroci (*switch*, ovvero invii) possono già essere controllabili da un elaboratore, che logicamente "automatizza" il loro uso memorizzando la loro posizione e le modifiche in tempo reale.
La connessione di questo tipo di mixer con un sequencer è logica perché è questa la funzione capace di generare il suo controllo. Comunque questo sequencer normalmente è un *sequencer dedicato* per due ragioni:

- perché funziona "permanentemente" in input/output, ovvero permette la registrazione-esecuzione sistematica ma non consente all'elaboratore di fare altro;
- perché le informazioni standard della codifica MIDI vengono adeguate in maniera particolare: ad esempio la nota non corrisponde logicamente alla nota se non ha un parametro di assegnazione diverso, così come altri eventi MIDI vengono interpretati a seconda delle necessità della funzione

da eseguire. Si perde il significato iniziale della codifica standard MIDI ma si sfruttano le sue "possibilità comunicazionali".

Esistono sul mercato software dedicati a questo scopo.
Una forma di automazione più ristretta rispetto a quella (potenziale) totale dei mixer via MIDI viene implementata in diversi mixer controllabili solo attraverso il *program change*; tutti i parametri interni di questi mixer MIDI in un momento determinato costituiscono un *programma.* La memoria massima permette la combinazione sequenziale di 127 *programmi* diversi.

2.8.14. *Rack d'effetti controllato via MIDI*

In questo caso la periferica risponderà con algoritmi di processo pre-codificati (*eco*, *delay*, *reverb* ecc.) assegnati a un valore di *program change.* Queste periferiche hanno normalmente la MIDI perché possano venire controllate dall'esterno in tempo reale per l'esecuzione dei loro diversi programmi, e anche in tempo differito per effettuare l'editing dei loro parametri attraverso appositi software di edizione tipo *patch editing.*
Comunque, con il sistema *disk-based digital recording* l'elaboratore può, mediante software, realizzare direttamente tutte le funzioni di queste periferiche sfruttando algoritmi di *post-production sound effects.*

Riassumendo, con la nostra workstation dotata di software adeguato e di qualche periferica hardware possiamo oggi controllare tutte le funzioni necessarie per la produzione di musica in uno studio "standard".

2.9. **Le famiglie di software per la MIDI**

La lettura del punto precedente ci ha fatto entrare in un universo di programmi capaci di rendere l'Atari ST il cuore di qualsiasi studio di registrazione professionale o semi-professionale.
Chiarito il concetto di home studio come rete locale e di elaboratore come workstation, vediamo adesso i programmi commerciali che rispondono alle diverse funzioni viste nello schema teorico precedente.

2.9.1. *Sequencer*

I sequencer sono, come abbiamo visto, i programmi di controllo della MIDI. I sequencer commercializzati fino ad ora sfruttano in modo ossessivo la metafora della multitraccia e questo è assai limitativo.
Ho visto convincere la gente della qualità di un determinato sequencer con argomenti come "permette 24.000 note", "ha 35 tracce". Le limitazioni di tracce nei sequencer sono dovute solo alla mancanza di immaginazione degli sviluppatori di software o alle limitazioni imposte da argomenti di carattere commerciale (poche tracce = prezzo basso; molte tracce = prezzo elevato: facciamo due prodotti, uno economico e l'altro a prezzo "professionale"). La memoria degli Atari ST permette tutte le tracce che si desiderino; limitarne il numero e considerare la traccia come una "unità basilare" è semplicemente assurdo.
L'insistenza sulla quantità di note nella pubblicità è perniciosa perché l'utente poco informato si lascia fuorviare a intendere il sequencer più come un programma per la composizione che come un programma gestore di informazioni MIDI.
Il numero di eventi MIDI che può essere memorizzato in un sequencer è proporzionale alla capacità di memoria della macchina usata (se quelli che sviluppano questi software conoscono il proprio mestiere, e a volte pare proprio di no); per essere affidabile, l'allocazione in memoria dell'informazione da eseguire deve essere *dinamica*; la macchina dovrà dunque sfruttare praticamente la maggior parte della sua capacità nella gestione dell'informazione e non bloccarla in memoria non operativa, perché quando un elaboratore lavora al massimo delle sue potenzialità di memoria è molto poco affidabile (si può bloccare in qualsiasi momento distruggendo tutta l'informazione processata e non salvata fino a quel momento) e anche nel migliore dei casi la sua velocità diminuisce logaritmicamente.
La metafora del multitraccia è limitativa perché obbliga a pensare a un solo livello di astrazione (quello delle singole sequenze).
Un sequencer evoluto dovrà sfruttare la possibilità di diverse pagine, dove le informazioni si trovano a diversi livelli, e soprattutto dovrà mostrare tutte le vere informazioni MIDI controllabili.
Ad esempio, nei sequencer è semplice registrare le sequenze di eventi tipo *key-on* o *key-off* e *key velocity* da tastiera; altre invece proprio non sono individuabili, però se vuoi organizzare il brano con una struttura in cui vuoi usare "pattern" prememorizzati — basso, accordi, drums — sono necessarie operazioni di ripetizione e trasformazione (il basso con lo stesso disegno ma

in relazione ad accordi diversi, nel rispetto dell'armonia), traslazioni di canali MIDI speculari (per effetti stereo), giochi di *pitch bend* e associazioni di effetti MIDI come *channel pressure* ecc.
La metafora diventa macchinosa, perché in realtà l'ambiente MIDI non è un multitraccia di sequenze di note: è molto di più.
Esistono programmi che sono "sequencer" pero che vengono *dedicati* a un uso speciale: il controllo di mixer MIDI. Questi programmi logicamente sono specifici del modello della periferica MIDI.

2.9.2. *Editori di suoni per campionatori*

Sono programmi che svolgono la funzione di *editing* di suoni digitali. Normalmente sono legati a un determinato campionatore perché le caratteristiche di formato dei campionatori non sono standard ma personalizzate.
Questi programmi sono venduti abbinati a una piccola libreria di *sound files* (fondamentalmente come esempio). Come già sappiamo i sound files (campioni memorizzati in un programma e non sul campionatore) occupano una gran quantità di memoria (dalla decina al centinaio di Kbyte).
Le librerie dei campionatori sono realizzate nel formato d'uso del campionatore, ossia con dischetti che possono venir letti dal campionatore e non dall'elaboratore; sono dunque "librerie di campioni" e non di *sound files.*
Un caso speciale di librerie di *sound files* sono le librerie su supporto CD-ROM: queste hanno bisogno di un software speciale che ne permette la trasmissione dal lettore di CD-ROM direttamente al campionatore. L'elaboratore in questo caso controlla la trasmisione.

2.9.3. *Editori di parametri*

Come succede con gli editori di suoni per campionatori, ogni modello di expander o sintetizzatore ha caratteristiche particolari di formato ecc.; dunque i programmi che svolgono questa funzione sono dedicati a un tipo specifico di periferica. In questo caso i programmi permettono:

- l'edizione del suono attraverso il controllo di specifici parametri;
- la configurazione di "banchi di preset" o "banchi di suoni" per trasmetterli via MIDI alla periferica o viceversa;
- l'organizzazione generale di una base di dati di suoni memorizzati come

file indipendenti o per gruppi di suoni o librerie (normalmente rispettando la capacità vera dei banchi fisici della periferica: ad esempio nel DX7 è di 32 suoni per libreria o file).

2.9.4. *Gli interfacciatori col codice SMPTE*

Sono programmi che permettono, attraverso l'uso di hardware aggiuntivo, ovvero di un'interfaccia speciale (interfaccia MIDI-SMPTE), la sincronizzazione tra sequencer e registratori che accettanno il codice SMPTE.
Queste interfacce, come già sappiamo, sono basilari per lavori di sonorizzazione di filmati o video.
Oggi il codice SMPTE abbinato al MIDI viene usato come sincronizzazione standard negli studi professionali, il che rende molto più sicura la sincronizzazione tra tutti gli elementi dello studio.

2.9.5. *Gli utility tools*

In informatica esistono programmi che servono per verificare e anche "riparare" malfunzionamenti software (formattazioni difettose di dischetti o hard disk, riconoscimento di periferiche ecc.). Questi programmi prendono il nome di *utility tools*, letteralmente "utensili di utilita" o, un po' più elegantemente, "programmi di servizio".
Anche per la MIDI esistono programmi analoghi. Uno dei programmi piu belli per MIDI si chiama *GenPatch ST MIDI Librarian / Utility* ed è stato scritto da John Eidsvoog.
Con questo programma, oltre ad avere un editore di parametri per una gran quantità di apparecchi (si chiamano editori "universali"), puoi controllare in qualsiasi momento il flusso dell'informazione MIDI dalla periferica al computer e anche dai floppy disk al computer e viceversa; puoi anche, attraverso "macro", riscrivere qualsiasi variazione su qualsiasi parametro. Si tratta comunque di un programma di uso complesso, perché per lavorare con codifica esadecimale bisogna essere utenti avanzati.

3

Computer music

La musica è una rivelazione più alta che la scienza o la filosofia.
L. V. Beethoven

Se hai problemi di composizione, orchestrazione, musicologia o semplicemente devi stampare le tue idee in maniera accettabile per la SIAE, questo è proprio il tuo capitolo. Ma non solo, in questo capitolo troverai sicuramente molto di più; troverai un po' di storia dell'informatica musicale o, come dicono gli americani, di "computer music".

Tutto un universo possibile grazie alla MIDI...
Nell'inverno 1985 Gareth Loy, un prestigioso commentarista di computer music scrisse un famoso "saggio" pubblicato sul numero 4 di quell'anno della più seria rivista americana del settore, il *Computer Music Journal*.
Il titolo era "Musicians Make a Standard: The MIDI Phenomenon". La traduzione è semplice: "I musicisti fanno uno standard: il fenomeno MIDI".
L'articolo considerava il fenomeno non a livello popolare ma nella prospettiva della musica "accademica" e anche dell'influenza dello standard sulle strade di ricerca dei musicisti informatizzati.
Le conclusioni non esageravano troppo: il MIDI è "inevitabile".
La parola "inevitabile" ha un accento pessimista, che nasce da un'analisi

severa della potenza di uno standard a livello di educazione di massa e anche a livello del rapporto interattivo, rispetto alla creatività umana.
Il MIDI è utile, però ha difetti, difetti che trovano una giustificazione nel campo storico-industriale nel quale è stato creato. Questi difetti sono di due tipi:

- fisici — sequenzialità e velocità di trasmissione;
- concettuali — natura e limitazione delle informazioni possibili nella codifica.

Ci sono campi di ricerca musicale nei quali questi difetti della MIDI rendono impossibile un suo impiego totale.
Due esempi caratteristici. Il controllo via MIDI di eventi musicali "simultanei" non è fisicamente possibile, perché in MIDI tutte le informazioni vanno una dietro l'altra, un accordo è sempre un arpeggio ad alta velocità e, quando eccede le otto note, la velocità di trasmissione dell'informazione sequenziale (che in termini informatici non è alta) provoca un fenomeno indesiderato: il tempo trascorso tra la prima e l'ottava nota è percepibile all'udito umano, dunque ci sono problemi apprezzabili di *delay* (ritardo).
Nel secondo caso i problemi sono ancora più complessi e per qualche aspetto non sono affatto superabili: è il caso dell'editoria musicale, che andiamo ad analizzare un po' più particolareggiatamente.

3.1. Perché è così difficile trovare un software per scrivere musica a partire dall'immissione dati via MIDI?

Uno dei campi più importanti, in cui la codifica MIDI ha generato ancor più confusione di quella già esistente, è stato quello della stampa musicale assistita dal computer o, come si dice tecnicamente, *editoria musicale elettronica.*
È notevole la quantità di informazioni, che sono necessarie per l'elaborazione della partitura in notazione, ma che i dati che costituiscono una "sequenza MIDI" non possono incorporare. Una sequenza MIDI non ha informazioni su tutti questi eventi necessari per la scrittura in notazione:

- le chiavi
- la divisione delle battute
- l'armatura ovvero la situazione di tonalità
- la misura del brano e le variazioni di una misura determinata inizialmente
- le differenze di alterazioni (diesis, bemolle, bequadro ecc.)

- l'esistenza di legature di durata o punti di valore
- l'esistenza di legature di portamento
- l'identificazione di valori di espressione in genere (staccato, tenuto ecc.)
- identificatori di abbellimenti
- informazioni grafiche necessarie (taglietti, gruppi irregolari)
- eccetera

Costruire un'approssimazione notazionale di una sequenza MIDI è difficile, e tante volte arbitrario (c'è sempre arbitrarietà nel generare informazioni che non possiedi).
Invece ci sono altre informazioni che una sequenza di eventi MIDI possiede inequivocabilmente, e sono quelle che permettono di estrarre informazioni significative per la notazione musicale:

- altezza precisa degli eventi musicali
- durata esatta degli eventi
- dinamica di ogni evento

Dunque, a partire da questi tre elementi conosciuti è necessario creare tutta l'informazione necessaria per la scrittura grafica in notazione, e questo si può fare (fino a un certo punto) cercando di eliminare al massimo le arbitrarietà (quelle comunque ci sono) e partendo dall'interpolazione dell'informazione esistente con una serie di *regole per difetto* via software.
I prodotti software che ne risultano sono molto complessi, per poter soddisfare anche al minimo le esigenze di registrazione per le società di proprietà intellettuale, e la complessità diventa ancora molto più grande se l'obiettivo è l'editoria.
Oggi infatti per il primo caso ci sono programmi con dei risultati abbastanza belli, che richiedono sempre un ritocco manuale, ma almeno risparmiano una notevole quantità di tempo. Per l'editoria non esiste nulla che permetta l'immissione diretta da una tastiera MIDI di *tutta* l'informazione (e qualsiasi pubblicità in questo senso mente).
Ci sono programmi che usano sistemi ibridi tipo "step input" riservando alle tastiere MIDI l'immissione dell'altezza dell'evento; il resto viene preselezionato da menu via mouse o tastiera alfanumerica.

I problemi fisici di sequenzialità della codifica normalmente vengono "compensati" via software, con processi chiamati di *quantizzazione*; cosi, ad esempio, un accordo di più note viene considerato un solo evento perché la

distanza equivalente alla sequenzialità del bus d'informazioni si corregge in modo automatico.
Questi processi di quantizzazione possono essere applicati non solo a questo livello (millisecondi), ma anche a intervalli di tempo maggiori, in modo da ricondurre la durata degli eventi (con correzioni arbitrarie per eccesso o per difetto) a una gamma di valori prescelti (ad esempio massimo valore accettato biscrome o crome o terzine di crome, ecc.)
Questi processi di correzione rendono almeno leggibili le informazioni musicali MIDI in notazione (sempre in modo approssimativo).
L'uso della registrazione con metronomo per musicisti esperti più l'applicazione della quantizzazione permettono un'immissione dati più veloce di quella step-input, però sono sempre necessarie correzioni, ed è ancora tutto da stabilire quale sia il metodo con cui si perde meno tempo.
Le informazioni relative alla dinamica interpretativa, equivalenti a *key on velocity*, sono di difficile valutazione; qualsiasi piccola variazione dinamica può corrispondere a fenomeni musicali diversi come accentazione, sforzando, staccato, tenuto, cambiamento immediato o progressivo d'intenzione dinamica, fraseggio ecc. Pertanto è preferibile evitarne l'automatizzazione di scrittura.

3.2. Un po' di storia dell'informatica musicale per capire meglio che cosa è realizzabile in un futuro imminente e che cosa tarderà ancora a realizzarsi

L'uso dell'elaboratore elettronico per applicazioni musicali è stato preso in considerazione fin dagli inizi dell'informatica.
La storia dell'informatica musicale nasce negli Stati Uniti: le prime ricerche risalgono al 1957 e furono condotte alla University of Illinois; ad esse si affiancano presto le ricerche ai Bell Telephone Laboratories, nel 1959, e successivamente alla Princeton University dal 1963.
Il 1965 è l'anno che vede l'Europa entrare in modo significativo nella storia dell'informatica musicale, con Utrecht, Pisa, Marsiglia, Londra come centri più attivi.
Negli anni sessanta, compositori di diverse nazionalità cominiciano ad usare l'elaboratore come ausilio nella composizione dei loro brani. In questa prima epoca si possono già segnalare diverse composizioni realizzate con l'ausilio di strumenti informatici: la *Illiac Suite* per quartetto d'archi di Lejaren Hiller e Leonard Isaacson (1957), *Atrees* di Ianis Xenakis (1962) *Übung für Klavier* di Gottfried Michael Koening (1969). Particolarmente belle sono le prime

trascrizioni di musiche di Bach e di Paganini realizzate da Pietro Grossi (1967) ai laboratori di ricerca della General Electric a Pregnana Milanese.
In questo periodo il musicista-compositore ha dovuto formalizzare, in modo più o meno consapevole, la propria particolare teorìa compositiva.
I musicisti "informatizzati" degli anni sessanta seguono in genere un'impostazione molto caratteristica: in questi inizi si servono di modelli di processi probabilistici, come metodi di generazione casuale e verifica rispetto a regole prefissate (metodo detto anche di Montecarlo, meglio noto per i giochi d'azzardo).
Invece negli anni settanta l'approccio è diventato più maturo, si è prodotto uno sviluppo più diversificato e generale, grazie senz'altro anche alla maggiore accessibilità e flessibilità dei sistemi di elaborazione.
In questi anni nascono i primi "linguaggi" per la composizione musicale mediante l'elaboratore. Il pioniere è Max Mathews che nei Laboratori della Bell Telephone negli anni 1969-1970 realizza il primo linguaggio chiamato MUSIC V (1969) e il sistema GROOVE (Mathews & Moore 1970).
Il MUSIC V, con i suoi derivati MUSIC IVBF, MUSIC 360, MUSIC 7, ecc., è il linguaggio compositivo più diffuso e conosciuto ancora oggi negli ambienti della computer music.
L'impostazione del MUSIC V è la simulazione delle apparecchiature elettroniche analogiche (oscillatori, filtri, generatori di forme d'onda e di inviluppo ecc.) con programmi per elaboratore che, in tempo differito, producono una sequenza di valori numerici; questi vengono trasformati a loro volta in un segnale analogico che può essere usato come ingresso per un comune amplificatore. La conversione è effettuata ovviamente attraverso un convertitore digitale/analogico, e questo metodo viene chiamato *sintesi digitale diretta*.
Con GROOVE, sviluppato da Max Mathews e Richard F. Moore, sempre ai Bell Telephone Laboratories (1973) l'approccio è ibrido, con l'obiettivo di evitare il tempo differito (tempo differito vuol dire che tra la "creazione" di un suono e il suo ascolto possono passare enormi quantità di tempo). GROOVE elabora numericamente il testo musicale e controlla digitalmente i parametri di apposite apparecchiature elettroniche capaci di generare suoni, che sono le periferiche del sistema (sintetizzatori ecc.), il che permette l'esecuzione e l'ascolto in tempo reale, cioè immediatamente.
Questi approcci degli informatici musicali teorici hanno caratterizzato la storia della computer music fino ad oggi.
Nel giro di una decina di anni, dalle idee sperimentali di Mathews e altri è scaturita la possibilità di rendere accessibile a livello domestico la creazione di musica di alta qualità sonora. Ovviamente grazie alla progressiva popola-

rizzazione degli elaboratori (che sono ormai diventati domestici o personali) e delle periferiche digitali (sintetizzatori, campionatori ecc).

3.2.1. *Computer music, o informatica musicale?*

Prima di proseguire, è opportuno tracciare con precisione una distinzione terminologica.
Dalle prime esperienze statunitensi è nato il termine *computer music*, che ben descrive l'atteggiamento con cui, prevalentemente, si sono accostati i ricercatori-musicisti americani all'elaboratore, per fare musica con l'elaboratore (automazione dell'interpretazione), per produrre testi musicali (prima) e suoni (poi, quando la tecnologia lo ha consentito).
In Europa l'atteggiamento è stato diverso, più generale, sono state considerate tutte le diverse possibili applicazioni dell'elaboratore in campo musicale: analisi, elaborazione e sintesi del testo musicale, analisi, elaborazione e sintesi del suono, stampe musicali, istruzione musicale assistita dall'elaboratore, ricerca musicologica assistita dall'elaboratore, ecc.
Questo diverso modo di avvicinarsi all'uso dell'elaboratore ha influenzato la definizione del nome della nuova disciplina: l'*informatica musicale.*
Naturalmente, anche negli Stati Uniti ci si occupa di tutti gli aspetti dell'informatica musicale, ma il termine che è rimasto per tradizione è quello più restrittivo, cioè computer music.

3.2.2. *L'informatica musicale come disciplina*

L'informatica musicale è la disciplina che si occupa dell'uso degli elaboratori per finalità musicali (e non della musica "fatta da un elaboratore"). È una disciplina molto giovane e la sua natura è ancora molto sfaccettata, poiché in essa rientrano tutte le attività di ricerca, di composizione, di interpretazione, di programmazione, di edizione, di istruzione, di esecuzione orientate all'analisi, alla trasformazione o alla sintesi di fenomeni musicali.
L'obiettivo primario di questa nuova disciplina consiste nel fornire al musicista, non necessariamente professionista, gli strumenti opportuni per esercitare attività musicali più concettuali, che permettano di rendere automatiche quelle attività che richiedono una manualità fine a se stessa o meglio ancora richiedono un intervento umano di tipo ripetitivo (non creativo).
Gli elaboratori ormai sono stati applicati a tantissimi aspetti, in rapporto con

la musica in modo diretto o indiretto. Perciò è necessario riflettere sull'uso di questo strumento (l'elaboratore) a scopi musicali e questo è l'atteggiamento "strategico" dell'informatica musicale.

3.2.3. *L'elaboratore è un nuovo "strumento musicale"*

L'elaboratore diventa un nuovo *strumento musicale* in senso lato, cioè uno strumento che consente di realizzare *qualunque* esperienza musicale; può sembrare un'affermazione molto forte e lo è, ma è anche vera: l'unico limite che si pone al musicista informatico è la capacità di comunicare alla macchina la conoscenza musicale necessaria per farle eseguire le proprie richieste.
L'elaboratore può effettuare processi di analisi, elaborazione e sintesi ai due livelli fondamentali del fenomeno musicale:

- testo musicale
- suono

Questi due livelli non sono scollegati fra loro, bensì interagiscono. La produzione musicale è fondamentalmente una attività di elaborazione di questi due aspetti, finalizzata alla realizzazione di musica.
Tra la partitura e il processo sonoro corrispondente si pone l'attività di esecuzione (o di interpretazione); l'elaboratore può sia eseguire, cioè produrre suoni esattamente corrispondenti a quanto specificato nella partitura, che interpretare, cioè eseguire quanto specificato nella partitura aggiungendo una quantità di informazione assente o elaborando tutta o in parte l'informazione musicale specificata nella partitura.

Lasciamo per il paragrafo 3.3 un approfondimento dei temi relativi all'analisi, all'elaborazione e alla sintesi del testo musicale, così come un esame degli aspetti di esecuzione non MIDI; qui voglio invece fermarmi sull'elaborazione numerica del suono, perché in un futuro prossimo non saranno più necessarie le periferiche per "generare suoni", ma basterà l'elaboratore.

3.2.4. *Elaborazione numerica del suono: le potenzialità dell'elaboratore come strumento*

L'elaboratore come già sappiamo è capace di trattare solamente informazioni

di tipo *discreto* ed è perciò necessario trasformare il suono in informazione discreta e viceversa per poter mettere in comunicazione il musicista e l'elaboratore mediante suoni.
La realizzazione di questa trasformazione consente poi all'elaboratore di agire sul suono per applicazioni di analisi, elaborazione e sintesi.

3.2.5. *Campionamento e quantizzazione*

La discretizzazione del suono viene realizzata attraverso due procedimenti:

- il *campionamento*
- la *quantizzazione.*

Il campionamento rende il suono discreto rispetto al tempo e consiste nel prelevare il valore di ampiezza del suono a intervalli costanti di tempo.
La quantizzazione rende il suono discreto rispetto all'ampiezza e consiste nell'associare a ogni valore di ampiezza, prelevato mediante campionamento, un numero espresso con una certa quantità di cifre binarie.
Più è frequente il prelievo di campioni e più è ampio l'intervallo di frequenze che possono essere acquisite o prodotte (*banda passante*); per poter descrivere in modo accurato un suono sono necessari molti campioni per ogni ciclo, ma il limite minimo di campioni per descrivere la frequenza del segnale audio è di due campioni per ciclo: da questo principio (detto *teorema del campionamento*) deriva il fatto che per avere una banda passante di frequenze audio percepibili di 20 kHz è necessario avere una *frequenza di campionamento* di almeno 40 kHz.
Quante più cifre binarie vengono usate per descrivere il valore di un campione, tanto più accurata è la quantizzazione; dall'accuratezza della quantizzazione dipende la gamma dinamica descrivibile nel segnale discreto; si calcola che ad ogni cifra binaria del campione corrisponda un incremento della gamma dinamica di circa 6 decibel (dB); la quantizzazione a 16 bit consente quindi una gamma dinamica di 96 dB, a 12 bit consente 72 dB, a 8 bit consente 48 dB, e così via.
È importante considerare che gli errori dei procedimenti digitali vengono introdotti essenzialmente proprio nelle trasformazioni dalla forma continua (analogica) alla forma discreta (digitale) e viceversa.
La quantizzazione pone in corrispondenza con tutti i valori di ampiezza compresi in un certo intervallo di ampiezza un'unica parola binaria; quanto più

è piccolo l'intervallo, tanto più è accurata la descrizione discreta; quanto più è grande l'intervallo, tanto più è grande l'imprecisione e quindi l'errore che viene introdotto (detto *rumore di quantizzazione*).
Il campionamento considera il valore di ampiezza del segnale analogico solo in certi istanti, a intervalli prefissati di tempo. Le variazioni del segnale all'interno dell'intervallo sono trascurate; se la frequenza di campionamento non è pari ad almeno il doppio della massima frequenza da campionare si verifica un fenomeno, detto *aliasing*, a causa del quale si producono erroneamente frequenze non presenti nel segnale originale o nel segnale discreto da sintetizzare.

3.2.6. *Le tecniche di sintesi*

Una tecnica di sintesi del suono consiste nel definire un modello formale capace di generare sequenze di campioni numerici (segnali audio digitali) a partire da parametri che variano lentamente nel tempo (con frequenza di variazione dei parametri inferiore alle frequenze audio).
I modelli noti in informatica musicale sono molteplici e tendono a privilegiare uno tra i due seguenti aspetti: la possibilità di ricostruire un timbro noto basata su una precedente fase di analisi o la possibilità di modificare dinamicamente un timbro in tempo reale (ad esempio durante una performance) mediante la variazione di un numero limitato di parametri.
Il primo aspetto è tipico delle applicazioni in studio, più accurate, inserite in un contesto di analisi-elaborazione-sintesi del suono, mentre il secondo è fortemente legato alle necessità dell'esecutore musicale che non può fisicamente controllare più che tanti parametri (così come un pianista non può suonare con più di dieci dita per premere i tasti dello strumento, anche il musicista informatico non può usare più di dieci dita per variare durante la performance i parametri del timbro).
Si è già visto infatti qualche esempio di esecuzione dal vivo di computer music dove l'interprete che "suona il timbro" non è un singolo individuo ma un gruppo di due e anche più persone a seconda del numero di parametri che variano in una certa tecnica di sintesi.

3.2.6.1. *La sintesi additiva*

La sintesi additiva si basa sul principio che una forma d'onda complessa può

essere ottenuta mediante sovrapposizione di un certo numero di forme d'onda semplici. Le forme d'onda semplici più utilizzate a tal fine sono forme sinusoidali. Generalmente le *formanti* (cioè le forme d'onda semplici) sono costituite da una parte che varia più lentamente relativa all'andamento dell'ampiezza della singola formante (*inviluppo parziale*) e da una parte che varia più velocemente, che descrive la forma ciclica della formante ovvero la frequenza; inoltre, un termine complessivo che varia lentamente descrive l'andamento globale dell'ampiezza (*inviluppo globale*).
La scelta di opportune forme d'onda semplici permette di generare qualunque forma d'onda per quanto complessa essa sia.
La sintesi mediante questa tecnica è in genere basata su una precedente fase di analisi, ovvero data una certa forma d'onda si calcolano i parametri che corrispondono alla sua scomposizione rispetto a un particolare insieme di forme d'onda semplici; a questo punto diventa possibile sintetizzare la forma d'onda di partenza assegnando al modello di sintesi i valori parametrici individuati durante la fase di analisi.
La sintesi additiva è forse quella che consente la maggior accuratezza e le maggiori possibilità di emulare i suoni naturali o in generale suoni già noti. Il suo uso è però sconsigliabile quando è richiesta un'efficiente operatività poiché rende necessaria la comunicazione dal musicista all'elaboratore di una notevole quantità di dati: un inviluppo di ampiezza e una fase per ogni singola formante più l'inviluppo globale; questo motivo rende la sintesi additiva inadatta per applicazioni in tempo reale.

3.2.6.2. *Table look-up*

Tra le tecniche più usate (specialmente per sintetizzatori digitali e campionatori) è la sintesi mediante consultazione di tabelle (detta anche *table look-up*). Questa tecnica prevede che i campioni della forma d'onda non debbano essere calcolati per ogni ciclo di sintesi, ma che sia memorizzato un intero ciclo della forma d'onda nella memoria dell'elaboratore o di un dispositivo specifico in forma tabellare; in questo modo si può leggere la tabella a diverse velocità di scansione (per generare diverse frequenze) o leggere all'interno della tabella con un certo "passo" (*sample increment*) cioè leggere un campione numerico e saltarne un certo numero in modo sistematico; entrambe le tecniche di scansione consentono il controllo della frequenza generata.
I limiti di questa tecnica di sintesi sono legati ai seguenti fattori: le dimensioni della tabella associata alla forma d'onda (se la tabella è di dimensioni modeste,

al crescere della frequenza la forma d'onda si snatura rapidamente fino a diventare un'onda quadra) e la staticità del timbro. Quest'ultimo problema può essere parzialmente ovviato utilizzando diverse tabelle in corrispondenza con i vari range frequenziali (tecnica di *multisampling*; ad esempio, una tabella per ogni ottava o per ogni singola nota).
La tecnica tabellare (detta anche *a forma d'onda fissa*) viene frequentemente usata per simulare (più o meno accuratamente) le timbriche degli strumenti musicali tradizionali sugli apparecchi per l'esecuzione musicale.

3.2.6.3. *Modulazione di frequenza*

Una tecnica che consente di controllare timbriche complesse mediante la variazione di un limitato numero di parametri è la modulazione di frequenza; essendo adatta per la programmazione del timbro in tempo reale ha conseguito molto successo tra i musicisti che suonano tastiere.
Il limite di questa tecnica è che mentre da una parte consente agevolmente di generare nuove timbriche complesse e dinamiche, dall'altra è uno strumento di lavoro poco apprezzabile per simulare timbriche di strumenti tradizionali; il motivo è molto intuitivo: non disponendo di strumenti di analisi semplici e affidabili come per l'analisi/sintesi additiva, di fronte al problema "quali parametri devo dare per avere quel certo timbro" la soluzione viene raggiunta mediante una grande esperienza nell'uso della tecnica e una lunga sessione sperimentale di prove acustiche.

3.2.6.4. *Altre tecniche di sintesi*

Molteplici sono le altre tecniche di sintesi studiate e applicate negli ultimi decenni; i più grossi sforzi della ricerca informatica e musicale vanno proprio nella direzione di cercare tecniche che permettano di controllare timbriche con evoluzione dinamica complessa controllabile mediante pochi parametri musicalmente significativi, tali da poter prevedere intuitivamente l'effetto che si può ottenere variando i parametri richiesti dalla particolare tecnica in oggetto.
Una tendenza generale recente è quella di studiare modelli più *fisici* che matematici, così da simulare il comportamento fisico degli strumenti tradizionali piuttosto che utilizzare un modello astratto che più difficilmente riesce a descrivere i diversi modi di "vibrare" dei corpi strumentali.

3.2.7. *Tecniche per la trasformazione del suono*

Una tecnica di trasformazione del suono consiste nel definire un modello formale capace di generare nuovi suoni mediante l'elaborazione di suoni precedentemente acquisiti o sintetizzati.
Tra le tecniche di trasformazione le più significative sono: la tecnica sottrattiva, la modulazione ad anello, la modulazione di ampiezza, il waveshaping, la riverberazione, il ritardo e i filtri.
Tralasciando per brevità le varie tecniche di trasformazione, accenniamo brevemente ai tipi di filtro più rilevanti: filtro *passa-basso* (attenua le componenti in frequenza al di sopra di una certa frequenza fissata), filtro *passa-alto* (attenua le frequenze inferiori ad una certa frequenza fissata), filtro *passa-banda* (attenua le frequenze esterne ad un certo intervallo di frequenze predeterminate), filtro *taglia-banda* (attenua le frequenze interne ad un certo intervallo di frequenze).

3.2.8. *Tecniche di analisi*

Una tecnica di analisi del suono consiste nel definire un modello formale capace di produrre, a partire dal suono, l'identificazione di parametri acustici.
Il modello di funzionamento dell'apparato percettivo umano riveste una rilevanza particolare nei procedimenti di analisi poiché i risultati di un'analisi sono meglio interpretabili se è facile riferirli al modo di sentire dell'orecchio; naturalmente, quando l'analisi è finalizzata all'estrazione di parametri legati ad una successiva applicazione di modelli di sintesi del suono diventa allora secondario il funzionamento dell'orecchio.
Ad esempio, la tecnica di sintesi additiva è generalmente basata su informazioni timbriche ottenute mediante tecniche di analisi che si intendono riprodurre o più semplicemente simulare.
Molti metodi di analisi sono legati all'applicazione della trasformata di Fourier e dei metodi di calcolo connessi.
Particolarmente critica è l'analisi nella fase dei transienti iniziali (detta anche fase di *attacco*) dei suoni, poiché la composizione spettrale varia molto rapidamente ed è tanto più difficile da analizzare quanto più varia in tempi confrontabili con le frequenze di campionamento.

3.2.9. *Registrazione digitale del suono*

Un settore in cui le tecniche digitali per il trattamento del suono hanno trovato ampia applicazione è quello della registrazione audio professionale.
Per questa attività si utilizzano sistemi in cui le capacità di elaborazione sono molto limitate, ma le unità di acquisizione (ADC) e di produzione (DAC) del suono sono di qualità molto elevata.
Il segnale audio analogico da registrare viene limitato in frequenza da un filtro passa-basso che limita la banda di frequenza alla metà (o meno) della frequenza di campionamento a cui opera il sistema di acquisizione; il nuovo segnale ottenuto viene campionato ed i campioni sono quindi convertiti in parole binarie (cioè quantizzati); a questo punto l'unità di elaborazione può effettuare tutte le operazioni di *editing digitale* sul segnale audio digitalizzato; ad esempio: filtraggio, riverberazione, compressione, amplificazione, eco, trasmissione a distanza, registrazione, playback, ecc.; quello che qualifica maggiormente la tecnica di registrazione digitale è che tutte le operazioni effettuate a livello numerico non producono degradazione della qualità del segnale. A questo punto viene compiuto il cammino inverso: le parole binarie sono convertite in tensioni (DAC) e quindi filtrate da un secondo filtro passa-basso che smussa i campioni analogici discreti prodotti dal convertitore digitale/analogico per ottenere un segnale audio analogico continuo.

Questa rassegna sull'elaborazione numerica del suono permette di immaginare che lo strumento musicale del futuro per applicazioni in tempo reale dovrà neccessariamente essere informatico e con una architettura disegnata appositamente: per questo ritorniamo un po' sull'idea di SEM (Sistema per l'Elaborazione Musicale).

3.3. **Lo stato dell'arte**

Gli elaboratori forniti di programmi e periferiche adatte, organizzati in modo razionale a scopi musicali, costituiscono un ambiente di lavoro definito sotto il concetto di SEM ovvero Sistema di Elaborazione Musicale. Le possibilità aperte dagli elaboratori modificano sostanzialmente il rapporto tra il musicista (creatore e interprete simultaneo in questo nuovo "ambiente") e il modo trazionale di creare (compositore - partitura - esecutore), offrono un aiuto anche al ricercatore musicale (musicologo, docente ecc.) negli aspetti meno rilevanti da un punto di vista concettuale (le azioni ripetitive fini a se stesse)

e propongono un nuovo atteggiamento rispetto a vecchi sistemi (analisi, insegnamento ecc.).

3.3.1. *Sistemi per l'Elaborazione Musicale (SEM)*

Fermiamoci sul concetto di SEM, che è un concetto fondamentale nell'informatica musicale, e rappresenta lo "strumento di lavoro" sempre più "totale".
La possibilità-necessità per il musicista di operare ai differenti livelli di rappresentazione modifica sostanzialmente il rapporto tra il musicista e il testo musicale: composizione ed esecuzione non sono più chiaramente distinguibili e la composizione è estesa fino al livello di composizione del timbro del suono in ogni sua caratteristica.
Quindi, piuttosto che di compositore/partitura/esecutore, diventa più opportuno parlare di *musicista/sistema per l'elaborazione musicale* (SEM), intendendo per SEM l'ambiente a disposizione del musicista.
La schematizzazione che distingue tra attività di analisi, elaborazione e sintesi dei testi musicali e dei suoni non è valida solo a livello concettuale o pragmatico, ma trova piena rispondenza anche nelle architetture e nelle tecnologie impiegate nei SEM; sono infatti ben differenziati i metodi e i dispositivi adatti per ognuna delle differenti attività.
Un SEM è composto da una serie di elementi hardware e da una certa quantità di software, organizzati in una "architettura": per esempio l'home studio MIDI visto precedentemente è una delle possibili "architetture" SEM. Vediamo ora in modo più dettagliato quali siano le architetture possibili, sotto un profilo concettuale.

3.3.1. *Architetture di SEM*

L'architettura di un SEM deve tener conto principalmente di due aspetti: l'*interazione musicista-SEM* e l'*elaborazione dell'informazione musicale* ai differenti livelli di rappresentazione.
In linea di massima si può considerare che per l'elaborazione di informazione musicale a livello strutturale, simbolico e fisico è sufficiente la tecnologia elettronica utilizzata per le applicazioni informatiche di tipo generale, mentre per le applicazioni che comprendono l'elaborazione a livello sonoro ed esecutivo sono necessari SEM interamente o parzialmente disegnati in modo

speciale, basati su tecnologie avanzate (a causa della grande quantità di dati da elaborare per unità di tempo).
I SEM possono essere classificati secondo i seguenti raggruppamenti in funzione dell'architettura adottata:

a) *SEM ibridi*: sono sistemi costituiti da un sottosistema digitale per le attività di livello più alto (strutturale, simbolico, fisico) e da un sottosistema analogico per le attività di livello più basso (fisico, esecutivo, sonoro); generalmente sono costituiti da un personal computer per l'analisi, l'elaborazione e la sintesi del testo musicale e da un sintetizzatore analogico per la sintesi del suono.
b) *SEM digitali ad analisi/sintesi diretta*: sono elaboratori di tipo generale attrezzati con convertitori analogico-digitali (ADC) per l'acquisizione del suono e con convertitori digitale-analogici (DAC) per la sintesi del suono; l'elaboratore è in grado di compiere attività a tutti i livelli di rappresentazione, ma solo in tempo differito poiché la tecnologia impiegata è troppo lenta per trattare in tempo reale il suono in forma numerica.
c) *SEM digitali misti*: questa soluzione prende in considerazione l'architettura multiprocessore nella sua forma più semplice; vengono associati due tipi di tecnologie diverse capaci insieme di gestire sia il testo musicale che il suono: per il testo viene usato un elaboratore di tipo generale di piccole o medie dimensioni, mentre per il suono viene usata tecnologia più avanzata, specializzata per la elaborazione numerica dell'informazione audio; nella sua configurazione tipo un sistema digitale misto prevede che l'elaboratore di più basso livello (suono) sia completamente asservito a quello di più alto livello (testo) che, ogniqualvolta sia richiesta una modifica di un parametro del suono, invia all'elaboratore "schiavo" le opportune istruzioni.
d) *SEM digitali multiprocessore*: estendendo la distribuzione dei compiti a più unità di calcolo e memorizzazione si ottengono i SEM multiprocessore; la filosofia di questo tipo di SEM è di associare un'unità di calcolo di opportuna tecnologia, con una adeguata quantità di memoria locale, ad ogni funzione del SEM; ad esempio: all'unità di acquisizione/produzione del suono, all'unità di elaborazione numerica del segnale audio, all'unità di scansione della tastiera musicale, all'unità di esecuzione degli algoritmi di analisi musicologica, ecc. Si distinguono due sottotipi:

- *SEM multiprocessore paralleli*
- *SEM multiprocessore gerarchici*

Nel primo caso i diversi processori comunicano tra di loro, ma non è stabilito a priori chi comanda. Nel secondo caso c'è un processore *master* che controlla gli altri (eventualmente organizzati a più livelli, cioè con processori che sono schiavi di un master e master a loro volta di altri processori di livello più basso).
I SEM digitali misti in pratica sono il caso più semplice di SEM multiprocessore gerarchico.

Si può ipotizzare che nel volgere di pochi anni i SEM avranno differenti tecnologie (sia per i processori che per le memorie) per le diverse funzioni musicali ai diversi livelli di rappresentazione.

Per meglio comprendere il significato di multiprocessore gerarchico o parallelo può servire una metafora musicale: si immagini che un processore sia un esecutore musicale; allora, il SEM gerarchico è un'orchestra con il direttore (il master) mentre il SEM parallelo è come un gruppo strumentale senza direttore in cui il coordinamento dei compiti avviene tramite i suoni stessi e tramite messaggi di varia natura.
Il SEM più generale sarà allora una combinazione di gerarchia e parallelismo, con un processore master al più alto livello che governa un gruppo di processori al livello inferiore a loro volta processori master per processori del livello inferiore e così via.

3.3.2. *Tecnologie e programmazione*

La soluzione di un problema mediante strumenti e metodi informatici passa sempre attraverso un opportuno dosaggio di capacità di calcolo, di memoria e di tecnologia più o meno veloce per rispettare i tempi dell'interazione tra uomo ed elaboratore.
Nel caso musicale si è già accennato come l'elaborazione del testo non abbia particolari esigenze, mentre l'elaborazione del suono richiede una notevole rapidità nei calcoli per poter operare in tempo reale.
Quando si vuole sviluppare un'applicazione in tempo reale è necessario allora usare tecnologia più veloce ed eventualmente, per ottenere un'ulteriore rapidità, porre in parallelo più unità di calcolo attive contemporaneamente.
Inoltre, poiché il calcolo della forma d'onda del suono è sempre molto oneroso, quando una forma d'onda deve essere utilizzata più volte è frequente la scelta di memorizzarla; in ogni caso è però necessaria una quantità considerevole di

memoria. Infine, deve essere possibile trasferire ad alta velocità grosse quantità di dati da un'unità funzionale del SEM all'altra.

Le tecnologie più avanzate permettono di realizzare circuiti VLSI (*Very Large Scale Integration*, cioè ad altissimo livello di integrazione, dove un componente può sostituire fino a un milione di componenti elementari) per l'elaborazione numerica dei segnali e Array Processor per l'elaborazione veloce di tabelle, in grado di soddisfare in tempo reale le esigenze di elaborazione di un SEM.
Per quanto riguarda l'elaborazione del testo musicale esistono almeno tre attività che fanno eccezione in termini di risorse necessarie: molto tempo è richiesto per l'esecuzione degli algoritmi di analisi musicologica (specie se polifonica), molta memoria è richiesta per costruire banche dati musicali o per effettuare stampe di partiture ad alta risoluzione grafica (nel caso si memorizzi la "mappa di bit" dell'intera partitura).
Per quanto riguarda le memorie, i supporti ottici e magneto-ottici consentono di avere a disposizione del SEM grossissime quantità di dati: si pensi che un dischetto del diametro di 12 cm (CD-ROM, *Compact-Disc Read-Only-Memory*) contiene fino a 600 milioni di byte e che un dischetto WORM (*Write-Once Read-Mostly*) contiene fino a 2600 milioni di byte.
La programmazione di questi dispositivi richiede linguaggi e tecniche specifici; in particolare, la programmazione di dispositivi con unità di calcolo operanti in parallelo richiede la capacità di controllare tutte le unità contemporaneamente in modo da sfruttare appieno le potenzialità del mezzo.
Le applicazioni di informatica musicale richiedono perciò lo sviluppo di software in cui si integrino porzioni di programmi ad alto livello (scritti con i linguaggi di programmazione comuni) con porzioni di programmi scritti con i linguaggi di basso livello tipici dei particolari dispositivi (generalmente microassembler).

Attualmente esistono tre categorie di SEM, classificate secondo il livello di programmabilità, cioè di quanto il musicista possa determinare il comportamento generale del SEM stesso:

- programmabili,
- programmati,
- strumenti musicali digitali.

Sistemi programmabili

Sono i sistemi più flessibili poiché tocca al musicista stesso sviluppare il software che li governa; in altre parole, sono sistemi "svestiti": a seconda del "vestito" che il musicista gli prepara possono fare un po' di tutto; gli unici limiti fisici sono dati dalla quantità di memoria a disposizione e dalla velocità di calcolo delle unità di elaborazione.
Questo genere di sistema viene generalmente utilizzato solamente nei centri di ricerca dove esiste personale preparato per la programmazione di elaboratori numerici di segnali audio; sono ancora assai rari i musicisti in grado di arrangiarsi da soli.

Sistemi programmati

Sono i sistemi commercializzati orientati al settore professionale degli studi di registrazione, delle performance di alto livello e anche dei centri di ricerca dove si produca musica elettronica. In questo caso, il SEM è già corredato del software musicale di base.
Il sistema è quindi meno flessibile, ma molto più semplice da usare; le funzionalità sono ridotte a quelle già programmate e generalmente molto orientate alla produzione di musica commerciale.
Nei sistemi più evoluti ci sono comunque buone possibilità specie di ricerca timbrica mediante programmi di analisi/sintesi del suono basati sulla scomposizione/composizione di suoni complessi in/con suoni semplici (tipicamente suoni a forma d'onda sinusoidale).

Strumenti musicali digitali

In questa fascia di SEM è più opportuno parlare di strumento musicale piuttosto che di sistema, trattandosi di apparecchi costruiti e programmati con caratteristiche fortemente orientate alla performance musicale.
La flessibilità è minima: sono disponibili, tipicamente, la tastiera musicale per suonare e qualche pulsante per impostare i parametri timbrici in accordo con la particolare tecnica di sintesi numerica implementata sull'apparecchio.
Mentre per i due tipi di SEM precedentemente discussi sono possibili applicazioni di analisi, elaborazione e sintesi del suono (e talvolta del testo), per questo tipo di SEM è possibile solo la sintesi del suono.
Gli strumenti musicali digitali sono a loro volta suddividibili in due tipi di strumenti diversi:

- i *sintetizzatori digitali* (che producono suoni sintetici mediante l'applicazione di opportuni modelli formali)
- i *campionatori* (che riproducono, eventualmente modificati o filtrati, suoni acquisiti mediante campionamento).

Che tipo di SEM era la rete dello home studio MIDI ?

Applicando le idee generali sulle architetture di SEM la rete vista e analizzata nel secondo capitolo, che è il tipo di rete più o meno standard proposta per l'industria musicale oggi, offre l'architettura di un *SEM multiprocessore gerarchico* dove l'Atari ST (che è un sistema programmato, però anche programmabile) è il master di diversi strumenti digitali o sistemi "dedicati" che sfrutta per le comunicazioni il protocollo MIDI.

3.3.3. *Un'idea in azione: Atari ST ed elaborazione/sintesi del suono in tempo reale!*

Come già sappiamo questa rete non può arrivare alla elaborazione e sintesi del suono in tempo reale.
Nel corso dello SMAU 87 a Milano, l'Atari, per bocca del progettista degli elaboratori ST e Mega, ha annunciato la presentazione sul mercato, verso l'estate del 1988, di una scheda "transputer" abbinata alle sue macchine.
I transputer sono un tipo di processori paralleli ad alta velocità: ogni transputer può elaborare quattro istruzioni alla volta ed essere collegato in catena con altri processori.
Questi processori, utilizzati fino ad ora a livello sperimentale e per applicazioni di grafica, offrono a livello tecnologico, con il software adeguato, la possibilità dell'elaborazione e sintesi del suono "in tempo reale".

3.3.4. *Applicazioni avanzate e futuribili nell'analisi elaborazione e sintesi del testo musicale*

Nel secondo capitolo abbiamo visto le operazioni realizzabili con il testo musicale con l'aiuto dell'elaboratore nella rete dell'home studio MIDI.
Vorrei qui approfondire gli aspetti più avanzati della ricerca attuale sul testo musicale nell'informatica musicale.

L'analisi, elaborazione e sintesi dell'informazione musicale significativa percorre oggi diverse strade.

L'analisi

L'analisi delle informazioni contenute in un testo musicale porta alla luce a livello musicologico moltissime cose. L'elaboratore è un aiuto ideale per la ricerca musicologica, perché può automatizzare gran parte del lavoro di routine: basta individuare il motivo della ricerca, immettere il materiale sorgente nell'elaboratore e progettare il software specifico.
Logicamente anche nel lavoro con gli elaboratori non mancano i problemi: ad esempio la ricerca incrociata è difficile, e deve essere prevista a livello di "data retrieval", cioè nell'organizzazione stessa delle basi di dati musicali.
La codifica d'immissione deve effettuare la traduzione dell'informazione in modo completo: questo è uno dei problemi più gravi, oggi, perché ancora non esiste uno standard per l'informazione musicale a livello di "testo musicale". Sono state formulate diverse proposte di standard (la più nota è la DARMS) ma nessuna gode ancora di una accettazione generale, e questo comporta una pesante limitazione alla generalizzazione dell'uso di programmi di analisi.
La realizzazione di programmi per l'analisi musicologica è un settore importante, così come per l'analisi linguistico-musicale (armonia tradizionale e non, contrappunto, forme musicali ecc.)

Elaborazione

La capacità umana di formalizzare (programmare) è l'unica limitazione all'uso degli elaboratori. L'elaborazione del testo musicale passa necessariamente attraverso la formalizzazione dei processi di trasformazione. Il compito più importante, oggi, per l'informatica musicale, è quindi quello di individuare i processi musicali più significativi per poterli formalizzare.

Sintesi

Le vie di ricerca sulla sintesi del testo musicale in informatica musicale vanno di pari passo con il resto dell'informatica: si sfruttano sempre più tecniche di generazione non algoritmica, attingendo alle possibilità offerte dall'Intelligenza Artificiale, dai linguaggi orientati agli oggetti (*object-oriented*), dalla sintesi mediante grafi di conoscenza (come le reti di Petri di cui parleremo più avanti) ecc.

3.3.6. *Informatica musicale e attività connesse al fenomeno musicale*

Il fenomeno musicale come cultura ha una storia universale, che è oggetto di studio della musicologia.
Il fenomeno musicale è presente in tutte le culture del mondo in modo caratteristico, e a diversi livelli di rappresentazione sociale: è questo l'oggetto di studio dell'etnomusicologia e della sociologia della musica.
Il fenomeno musicale come fenomeno della comunicazione umana non verbale (estetica) è parte importante dell'oggetto di studio delle discipline che si occupano di questi fenomeni (scienze della comunicazione e dei mass-media, teoria della percezione, teoria dei linguaggi, filosofia, psicologia, teoria dello spettacolo ecc.).
Il fenomeno musicale, infine, è anche un'attività umana che permette la fruizione (fondamentalmente è un'attività ludica) e gran parte del suo impatto sociale dipende da questo.
Lo studio e il perfezionamento dei diversi atteggiamenti di fruizione del fenomeno (teoria musicale, tecnica strumentistica, uso della voce, espressione del corpo abbinata alla musica ecc.) ha invece in ogni cultura una tradizione complessa e spesso agonistica (si pensi per esempio alle scuole di virtuosismo strumentale).
Come in ogni attività umana, il suo "insegnamento", ovvero la trasmissione culturale della conoscenza precedente, ha metodi e tecniche precise e in costante rinnovamento.
In tutti questi campi oggi l'elaboratore è presente, in un modo o nell'altro, e l'informatica musicale, di conseguenza, è impegnata nello studio di queste particolari applicazioni.
I settori più importanti, sul piano dell'informatica, collegati a questi aspetti collaterali del fenomeno musicale sono:

- Desktop publishing musicale o editoria musicale elettronica
- Basi di dati musicali per la ricerca musicologica
- CAI o CBMI (*Computer-Based Music Instruction*)
- Ricerca formale-linguistica
- Software gestionale specifico per gli ambienti professionali legati alla musica
- Software per la post-produzione audio (finalizzato a performance teatrali o all'immagine)

... e tanti altri ancora.

3.4. Le reti di Petri: una rappresentazione formale per la conoscenza musicale

Le reti di Petri sono un grafo formale particolarmente adatto alla rappresentazione della conoscenza umana e alla rappresentazione di processi in generale. Le reti sono nate grazie ai lavori di ricerca logico-formale di Carl Adam Petri (GMB di Bonn, in Germania). e formalizzate in un testo storico dal titolo *General Net Theory* (1976). Il loro uso per applicazioni di carattere musicale data solo dagli inizi degli anni ottanta (grazie a lavori realizzati da G. Degli Antoni e G. Haus nel 1982, poi da Camurri, Haus, Zaccaria nel 1985 e da Stefan Pope nel 1986).
Le reti di Petri (RdP) sono di semplice uso e si possono imparare rapidamente: l'atteggiamento giusto per imparare a usarle è immaginare che si tratti di un gioco con regole inizialmente semplici, come ad esempio il gioco degli scacchi. Giocare bene poi dipende dalla capacità di generare metafore e strategìe (il che vale anche per la composizione musicale).

Gli elementi

Le RdP hanno solo quattro simboli:

- *Posti* (*Place*) = che sono rappresentati con cerchi
- *Transizioni* (*Transition*) = che sono rappresentate con tratti o rettangolini
- *Archi orientati* (*Oriented arcs*) = che con l'orientamento della freccetta indicano la direzione del processo
- *Marche* (*Tokens*) = che sono rappresentate o da pallini o da numeri interi positivi e si possono trovare solo entro i posti.

Nella figura 3.1 vediamo questi elementi in una rete elementare.
Una rete si ottiene collegando posti e transizioni sia in ingresso che in uscita. La prima regola da memorizzare è che in una RdP *non si possono collegare posti con posti e transizioni con transizioni.* E questo perche?
La metafora usata in queste reti porta ad associare al circolo il concetto di risorsa e alla transizione il concetto di "attività".
I posti connessi in ingresso a una transizione rappresentano le "risorse necessarie perché si possa produrre quell'attività", i posti connessi in uscita a una transizione rappresentano "le risorse prodotte da quella attività", i posti connessi in ingresso e in uscita a una transizione sono "risorse che quell'attività non consuma" (come i catalizzatori in una reazione chimica).

Marche
Posto
1
Archi orientati
Transizioni
Descrizione Statica
Posto in ingresso
Marcatura iniziale
Posto in uscita
Descrizione Dinamica
scatto di una
transizione
raggiungibilità
di una marcatura

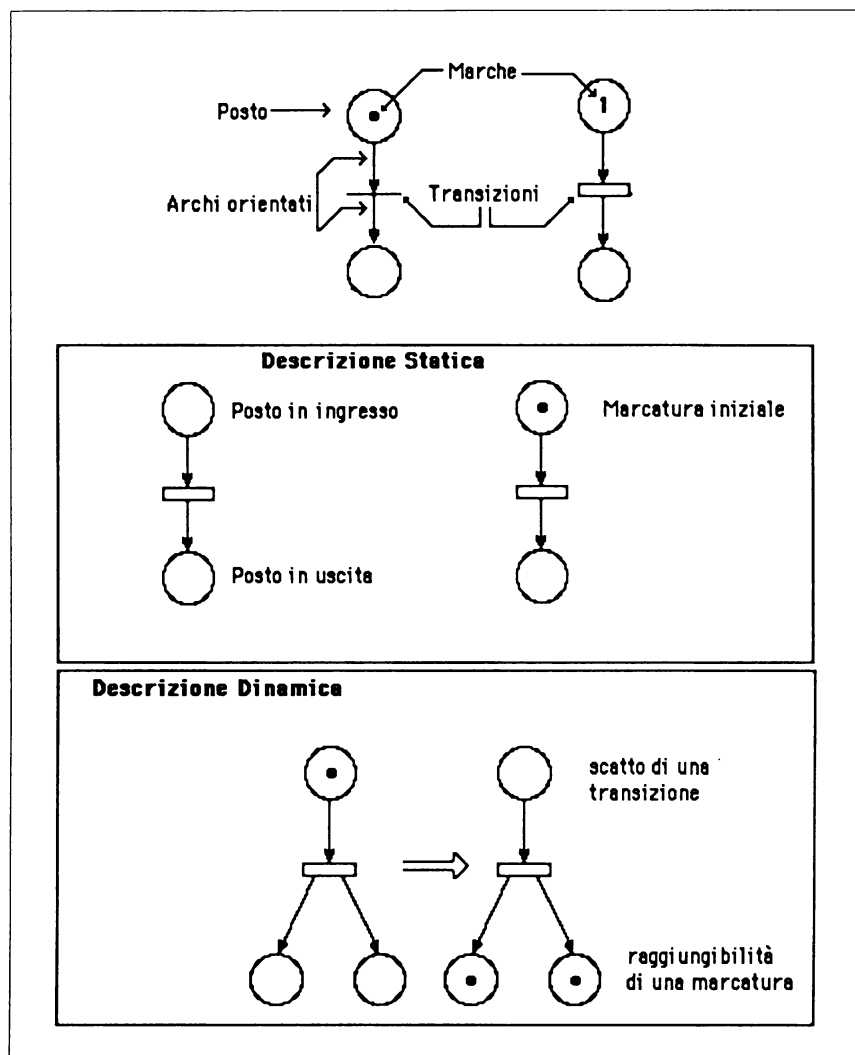



Fig. 3.1

Se si collegano attività con attività o risorse a risorse la metafora perde il suo senso.

Possono esistere attività che consumano e che non producono niente? Naturalmente. La loro rappresentazione in rete è una transizione con solo posti in ingresso e senza posti in uscita.

Possono esistere attività che non consumano ma che producono ?
Magari. Comunque a livello di rappresentazione verrebbero rappresentate come transizioni senza posti in ingresso ma con posti in uscita.

3.4.1. *Le definizioni*

a) Descrizione statica
A livello di definizioni formali: una rete di Petri (RdP) è un *grafo* in cui si possono riconoscere due tipi di nodi: i *posti*, rappresentati mediante cerchi e le *transizioni* rappresentate mediante tratti.
I nodi possono essere *conessi da archi orientati* purché i nodi che si congiungono siano di tipo diverso; sono possibili connessioni dai posti alle transizioni e dalle transizioni ai posti, mentre non sono ammessi archi congiungenti posti con posti o transizioni con transizioni. *I nodi di tipo posto possono contenere marche*, rappresentate mediante puntini ingrossati o numeri interi positivi.
Si dice che un *posto è in ingresso* ad una transizione quando è connesso alla transizione da un arco orientato dal posto alla transizione.
Viceversa, si dice che un *posto è in uscita* da una transizione quando è connesso alla transizione da un arco orientato dalla transizione al posto.
Si dice *marcatura iniziale* di una rete di Petri (RdP) la disposizione di marche nei posti della rete in un certo istante, detto appunto iniziale.

Una rete è descritta completamente quando sono noti:

- l'insieme dei posti della rete
- l'insieme delle transizioni della rete
- le connessioni dei posti in ingresso alle transizione della rete
- le connessioni dei posti in uscita alle transizioni della rete
- la marcatura iniziale della rete.

Queste informazioni danno la descrizione statica della rete.

b) Descrizione dinamica
Le reti di Petri però non sono solo una descrizione statica di un processo (come lo sono la partitura notazionale e altre forme grafiche di rappresentazione degli oggetti musicali); ciò che le rende particolarmente attraenti come grafi formali è che permettono la descrizione dinamica di processi.
Questo significa che le reti possono modificarsi ed evolvere. Le RdP non sono solo una descrizione statica di un processo, anche dinamico, ma proprio una descrizione dinamica e causale di processi deterministici o non. Il che offre delle potenzialità descrittive finora insuperate per la descrizione di processi musicali.
Vediamo come possono dare una descrizione dinamica.

Alle reti di Petri è possibile associare la descrizione della loro evoluzione, cioè delle variazioni della marcatura della rete.
Le variazioni della marcatura (*marking*) si hanno quando si verificano *scatti (firing) di transizioni.*

Si dice che una transizione può scattare quando tutti i posti in ingresso alla transizione hanno almeno una marca. Il momento in cui avviene lo scatto non è determinato; l'evoluzione temporale di una rete non è perciò definita. Sono invece definite le relazioni logico-causali.
L'effetto di uno scatto di una transizione è di:

- eliminare una marca da ogni posto in ingresso alla transizione
- aggiungere una marca ad ogni posto in uscita alla transizione.

Si dice, infine, che una marcatura è *raggiungibile* quando esiste almeno una successione di scatti di transizioni della rete che, a partire dalla marcatura iniziale della rete, porta alla marcatura in oggetto.

Usando la metafora iniziale, la marca dentro una risorsa indica la sua "disponibilità"; una determinata attività si potrà realizzare se e solo se tutte le risorse necessarie (posti in ingresso) per detta attività sono "disponibili" (hanno almeno una marca); la realizzazione dell'attività (scatto della transizione) produrrà la disponibilità delle risorse prodotte da essa (aggiungendo una marca a tutti i posti in uscita).

Dunque la seconda regola del gioco è: che ogni transizione (trattini o rettangolini) scatterà quando tutti i posti in ingresso (cerchi con archi orientati verso

la transizione) hanno almeno una marca (puntino o numero intero positivo). Il risultato dello scatto della transizione produrrà una marca in tutti i posti in uscita (un puntino o l'uno in tutti i cerchi con archi orientati dalla transizione). Il momento nel quale si produrrà lo scatto non viene determinato: in questa prima approssimazione interessa solo conoscere il rapporto causale-logico, e non simulare la temporizzazione di un determinato processo (questo comunque è possibile con una particulare estensione delle RdP, le *reti di Petri temporizzate*).

3.4.2. *RdP musicali*

Vediamo adesso la particolare metafora che permette di applicare le RdP alla musica. I posti possono rappresentare due tipi di risorse:

- oggetti musicali osservabili (a qualsiasi livello di astrazione)
- oggetti di controllo (oggetti privi di una funzione musicale diretta però necessari per il controllo logico-causale dei processi rappresentati dalla rete)

Le transizioni rappresentano trasformazioni.
Con questa metafora un posto in ingresso rappresenta l'oggetto musicale o di controllo necessario per una data trasformazione, lo scatto della transizione significa la realizzazione della trasformazione che come risultato produce oggetti musicali o di controllo (marche nei posti in uscita alla transizione).

3.4.3. *Qualche rete basilare per la rappresentazione di fenomeni musicali*

Partendo da un solo oggetto musicale sono possibili tre costrutti di reti molto significative.

1. *Sequenza* (*sequence*) Significa che il processo di trasformazione di un oggetto musicale ne genera un altro; a livello più astratto questa rete serve per descrivere il dettaglio di un oggetto musicale che procede sequenzialmente come una successione di oggetti musicali nel tempo; senza dubbio è la rete più banale, se tutti gli oggetti musicali fossero sequenziali l'uso delle RdP non sarebbe necessario, perché la sequenzialità viene spiegata benissimo con i diagrammi di flusso e con altri tipi di grafi meno evoluti.

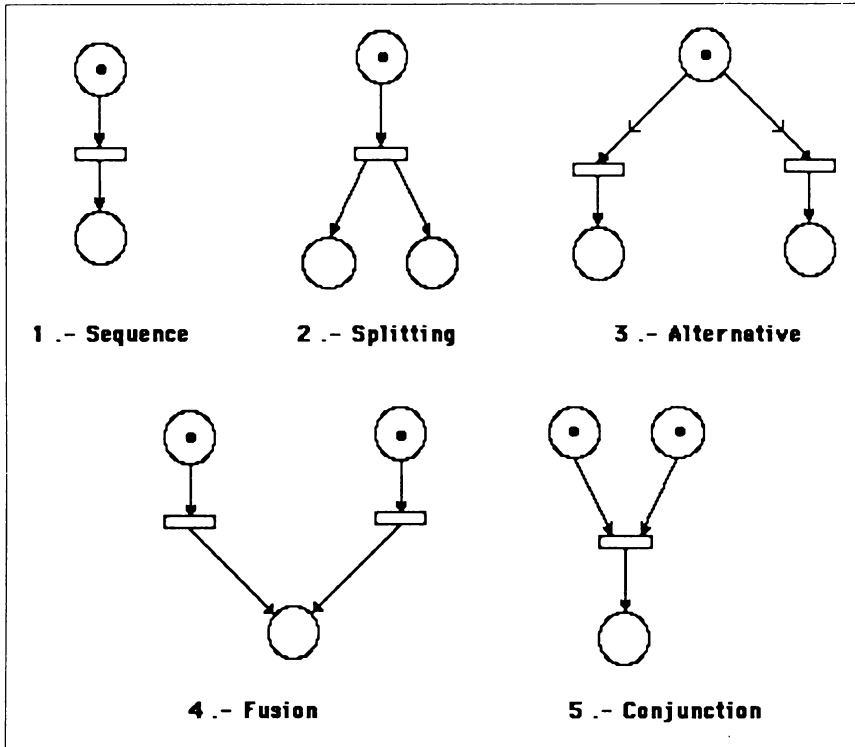



Fig. 3.2 Reti fondamentali per la rappresentazione di fenomeni musicali: nelle 1, 2 e 3 si parte da un solo oggetto musicale, nelle 4 e 5 si parte da più oggetti.

2. *Separazione* (*splitting*) Questo costrutto significa che da un solo oggetto musicale una sola trasformazione può generare due (o più) nuovi oggetti diversi.
Una tecnica musicale che sfrutta la separazione è l'armonizzazione a diverse voci di una melodia come oggetto musicale iniziale. Fenomeni di "splitting" in musica sono fondamentali nella composizione, orchestrazione ecc. Anche questo costrutto si può rappresentare con altri formalismi, sia pure in modo meno immediato.

3. *Alternativa* (*alternative*) L'alternativa invece non è rappresentabile con alcun altro formalismo musicale; invece esiste in musica. È la rappresentazi-

one, prima di scattare (detto in parole povere), del dubbio.
Analizziamo più in dettaglio il costrutto. L'oggetto musicale è disponibile per una sola trasformazione, ma è in ingresso a due possibili trasformazioni diverse; pertanto, se se ne realizza una, l'altra non si realizza e viceversa. Questo costrutto rappresenta un rapporto causale ma non deterministico.
Ad esempio, un compositore si può trovare con più di una "alternativa" a disposizione, e per finire il brano dovrà scegliere; nel caso in cui voglia lasciare la scelta all'interprete non ha un mezzo formale convenzionale per comunicare la sua decisione in modo non equivoco. Con le RdP invece abbiamo la possibilità di rappresentare formalmente il non determinismo.

Partendo da piu di un oggetto musicale ci sono due costrutti significativi:

4. *Fusione* (*fusion*) Un unico processo di trasformazione che a partire da due oggetti musicali in ingresso genera un unico oggetto in uscita.
Un esempio musicale tipico è il missaggio.
Questo costrutto e questo esempio servono per indicare che con le RdP i fenomeni musicali formalizzabili non sono solo quelli della musica tradizionale o classica: le RdP sono adatte per formalizzare qualsiasi tipo di fenomeno musicale di qualsiasi cultura a qualsiasi livello, permettono per esempio di formalizzare il lavoro di produzione musicale in studio, in modo da ottimizzare i passi.

5. *Congiunzione* (*conjunction*) Questo costrutto descrive diversi oggetti musicali che vengono trasformati da processi diversi, generando un unico oggetto musicale risultante.
È senza dubbio un costrutto normalissimo a livello musicale; qualsiasi elaborazione di un brano "a parti" è una congiunzione, ad esempio in un brano orchestrale suonato dal vivo da diversi musicisti (considerando in questo caso la performance come trasformazione), ogni musicista realizza il suo oggetto musicale, la congiunzione di tutti è l'oggetto musicale complessivo che ascolta il pubblico.
Un altro esempio è la composizione strutturale ove si costruisce il brano a partire dall'accoppiamento (in senso logico) di diversi oggetti musicali primitivi e diversi processi a essi collegati.

Vediamo adesso l'uso degli oggetti di controllo associati alla rete con due esempi tipici di fenomeni musicali.

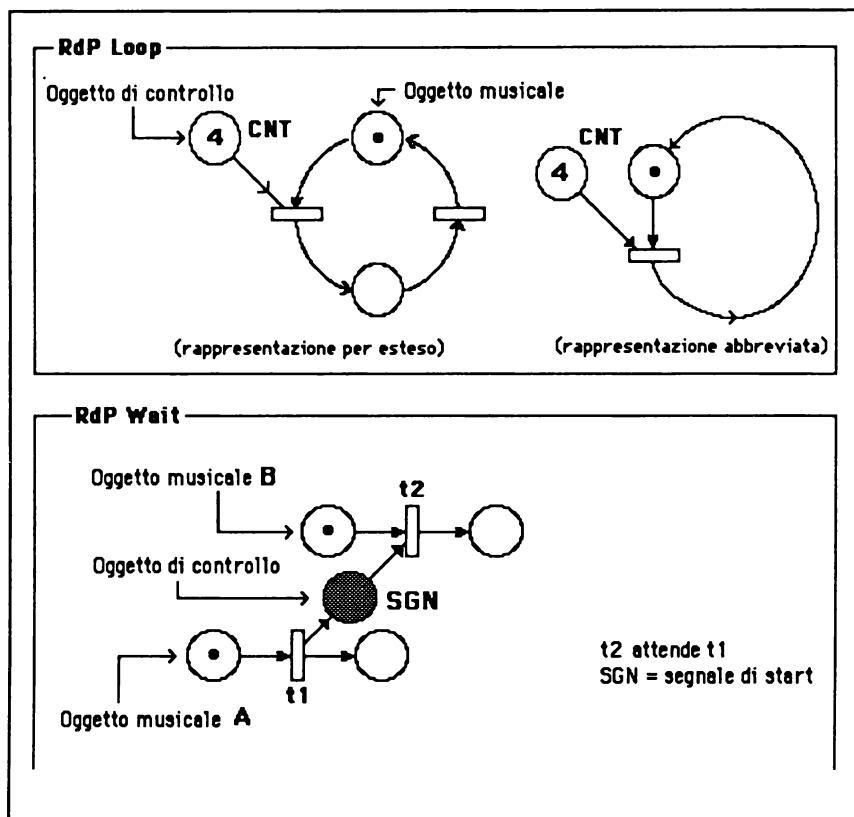



Fig. 3.3

Rete Loop: in questa rete l'oggetto di controllo chiamato CNT(contatore) ha come funzione "contare" il numero di volte che deve essere ripetuto l'oggetto musicale (4 nel caso specifico della figura). Il costrutto può essere rappresentato per esteso o in modo abbreviato (gli archi bidirezionali nelle RdP sono un'estensione semantica riassuntiva).
Con questo tipo di rete si possono rappresentare fenomeni musicali di repetizione dove l'oggetto musicale è normalmente un "pattern" e l'oggetto di controllo un contatore; ogni volta che si verifica uno scatto il processo consuma una marca del posto di controllo, e la ripetizione si blocca quando il posto di controllo rimane senza marche (nel nostro esempio alla quarta volta).

Rete Wait: Questa rete è un costrutto logico di priorità assoluta: il processo t1 dovrà concludersi perché si possa produrre il processo t2, dunque il secondo processo si trova in situazione di "attesa" o "wait" del primo. Il modo in cui il secondo processo riceve l'ordine di "start" è attraverso l'oggetto di controllo SGN (segnale di start).
Le applicazioni musicali di queste tipo rete sono molte: ad esempio, permettono di avviare processi collaterali al processo centrale in un momento preciso. Lo scatto della transizione t1 genera un oggetto musicale e una marca nel posto SGN, e la presenza di una marca in questo posto permette lo scatto della transizione t2 che fino a quel momento era in situazione di attesa per mancanza di marche in tutti i posti in ingresso.

Operatori musicali

Sono costrutti che servono per mettere in relazione processi diversi in modo formalmente non equivoco:

Concatenatore (*linker*): Permette di concatenare due oggetti musicali il secondo oggetto come proseguimento del primo; permette, in altre parole, di costruire sequenze.
Disgiuntore (*unlinker*): Ha la funzione opposta, cioè scindere sequenze in sottosequenze.
Sincronizzatore (*synchronizer*): Ha il compito di unire due processi musicali che hanno luogo simultaneamente, sincronizzandoli mediante transizioni.
Desincronizzatore (*de-synchronizer*): Serve a rendere indipendenti temporalmente (o meglio, causalmente) due processi musicali simultanei.

Fino qui le potenzialità descrittive delle RdP sono interessanti e in qualche aspetto superano le altre potenziali notazioni, però quello che le fa non solo interessanti ma uniche è che offrono la possibilità di morfismi.
Questo al musicista che non è abituato (in modo consapevole) a lavorare con formalismi matematici può non dire niente, quindi approfondiamo un po' questo aspetto.

Il *morfismo* consiste in una trasformazione di una rete di Petri in una rete di Petri differente secondo un certo criterio (ad esempio di astrazione o di dettaglio). Questo permette sia procedimenti *bottom-up* (astrazione) che *top-down* (dettaglio).
Detto in altre parole, con i morfismi si possono rappresentare intere reti

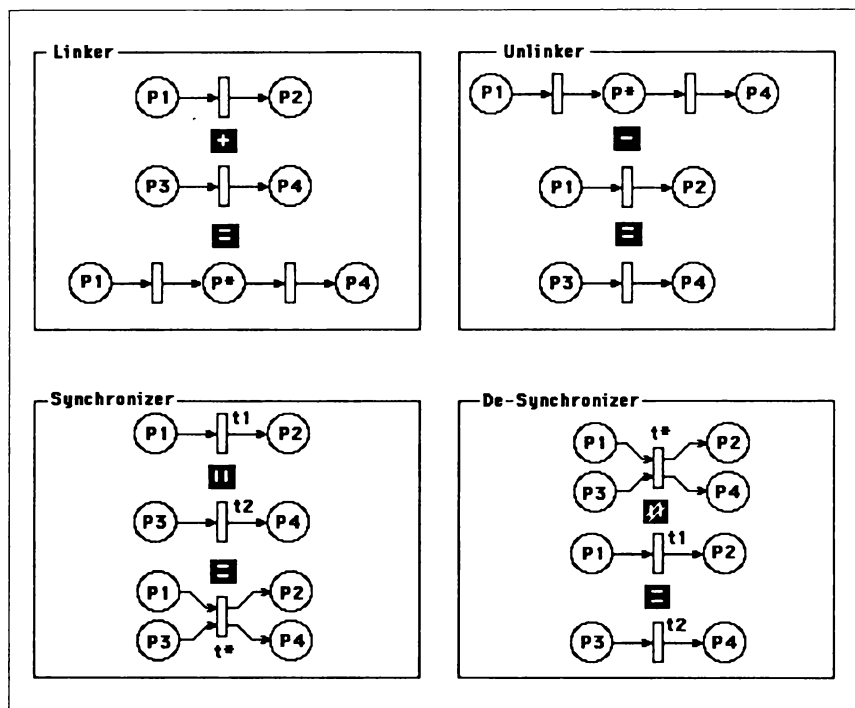



Fig. 3.4 Gli operatori musicali.

dettagliate con una sola transizione (o un solo posto) di una rete di livello superiore e viceversa. Il che permette l'uso di un *unico* strumento formale di rappresentazione a qualsiasi livello di astrazione.
Questo fino adesso non era possibile con nessun tipo di strumento formale di descrizione di fenomeni musicali. In notazione tradizionale, ogni livello ha una codifica diversa (e qualche livello non l'ha).

3.4.4. *RdP come interfaccia utente*

Le Rdp sono nate in ambiente informatico ed hanno la loro applicazione abituale nella formalizzazione di analisi e specifiche in ingegneria del software.

Le RdP (con opportuni programmi esecutori) possono diventare non solo un formalismo di analisi ma anche di elaborazione e controllo. In informatica musicale il loro uso può fornire al musicista un'interfaccia formale verso l'elaboratore.
L'uso di programmi essecutori di reti di Petri può risparmiare al musicista di imparare linguaggi di programmazione per controllare "totalmente" l'elaboratore.

3.4.5. *Esecutori di RdP come "controller" di SEM evoluti*

Le RdP sono il formalismo adatto alla rappresentazione (programmazione) di processi "concorrenti", e i SEM futuribili con multiprocessori ad alto parallelismo sono un ambiente hardware particolarmente adatto alla realizzazione in tempo reale di processi controllati da esecutori di RdP.
In questo senso sono particolarmente interessanti i lavori di ricerca e sviluppo di esecutori di RdP per musica realizzati al Laboratorio di informatica musicale del Dipartimento di Scienze dell'informazione dell'Università degli Studi di Milano (Camurri, Haus, Jacomini 1985; Haus, Bianchi 1986; Camurri, Haus, Rodriguez 1987) e ai laboratori della Xerox a Palo Alto negli USA da S. Pope (1987).

4

Atari ST e utenti avanzati a livello informatico

La musica è un esercizio di aritmetica segreta, e quelli che operano con essa ignorano che lo fanno con i numeri.

G. W. Leibnitz

Gli utenti con conoscenze già evolute nel campo dell'informatica possono trovare negli Atari ST e Mega elaboratori economici e potenti; questo capitolo si rivolge a loro e vuole essere una guida agli utensili per lo sviluppo software su queste macchine.

4.1. **Anatomia degli ST**

Gli Atari ST sono elaboratori general-purpose costruiti intorno al chip Motorola MC 68000 16/32 bit come CPU (*Central Processing Unit*). Il 68000 è poi contornato da diversi chip tanto "off-the-shelf" come "esclusivi".
Il sitema operativo degli ST, il TOS™, consiste di diversi moduli a sé stanti, il cui uso permette tanto di elaborare l'informazione come di controllare le diverse funzioni di I/O con le periferiche. Questi moduli sono:

• BIOS (*Basic Input/Output System*):
Il BIOS degli ST fornisce la gestione a basso livello dei servizi di I/O con la

<table>
<tr><th></th><th>520ST</th><th>1040ST</th><th>Mega2ST</th><th>Mega4ST</th></tr>
<tr><td>Processore</td><td colspan="4">Motorola MC 68000</td></tr>
<tr><td>architettura interna</td><td colspan="4">32 bit</td></tr>
<tr><td>architettura 'sterna</td><td colspan="4">16 bit</td></tr>
<tr><td>frequenza di clock</td><td colspan="4">8 MHz</td></tr>
<tr><td>Caratteristiche di sistema:</td><td colspan="4"></td></tr>
<tr><td>registri per dati</td><td colspan="4">8 da 32 bit</td></tr>
<tr><td>registri per indirizzi</td><td colspan="4">9 da 32 bit</td></tr>
<tr><td>bus dati</td><td colspan="4">a 16 bit</td></tr>
<tr><td>bus indirizzi</td><td colspan="4">a 24 bit</td></tr>
<tr><td>livelli di interrupt</td><td colspan="4">7</td></tr>
<tr><td>comandi</td><td colspan="4">56</td></tr>
<tr><td>modi di indirizzamento</td><td colspan="4">14</td></tr>
<tr><td>tipi di dati</td><td colspan="4">5</td></tr>
<tr><td>Memoria:</td><td colspan="4"></td></tr>
<tr><td>RAM</td><td>520 Kb</td><td>1040 Kb</td><td>2 Mb</td><td>4 Mb</td></tr>
<tr><td>ROM</td><td colspan="4">196.608 b</td></tr>
<tr><td>Risoluzione grafica</td><td colspan="4">selezionabile: 640 × 400 monocromatica
320 × 200 × 16 colori
640 × 200 × 4 colori</td></tr>
<tr><td>Colore</td><td colspan="4">tavolozza da 512 colori</td></tr>
<tr><td>Interfacce:</td><td colspan="4"></td></tr>
<tr><td>porte MIDI In, MIDI Out</td><td colspan="4">sì</td></tr>
<tr><td>porte monitor</td><td colspan="4">sì</td></tr>
<tr><td>porta parallela</td><td colspan="4">sì</td></tr>
<tr><td>porta seriale RS232</td><td colspan="4">sì</td></tr>
<tr><td>porta floppy disk</td><td colspan="4">sì</td></tr>
<tr><td>porta hard disk</td><td colspan="4">sì</td></tr>
<tr><td>porta cartucce ROM</td><td colspan="4">sì, per cartucce da 128 Kb</td></tr>
<tr><td>porte mouse/joystick e joystick</td><td colspan="4">sì</td></tr>
<tr><td>porta TV</td><td colspan="4">opzionale</td></tr>
<tr><td>Generatore di suono</td><td colspan="4">chip YM 2149</td></tr>
<tr><td>Tastiera</td><td colspan="4">intelligente di 94 tasti con microprocessore 6301</td></tr>
<tr><td>Disk drive:</td><td colspan="4"></td></tr>
<tr><td>densità di tracce</td><td colspan="4">135 tracce per pollice</td></tr>
<tr><td>capacità</td><td>360 Kb</td><td>720 Kb</td><td>720 Kb</td><td>720 Kb</td></tr>
<tr><td>supporto</td><td colspan="4">dischetti da 3.5"</td></tr>
<tr><td>velocità di trasferimento dati</td><td colspan="4">250 kilobit al secondo</td></tr>
</table>

Tab. 4.1 Le caratteristiche degli Atari ST e Mega.

tastiera, lo schermo e le porte di comunicazione con le periferiche.

• XBIOS (*Extended BIOS*):
Lo XBIOS fornisce lo stesso tipo di servizi che il BIOS, però più orientato verso funzioni di controllo di particolari periferiche hardware accettate dagli ST ma non dagli standard general-purpose, come il controllo delle porte MIDI, il chip interno di suono o i comandi per i disk controller ecc.
D'altra parte lo XBIOS fornisce anche un ventaglio di funzioni che permettono la manutenzione del clock del sistema e la commutazione del processore in *supervisor mode.*

• GEMDOS (*GEM Disk Operating System*):
È un DOS relativamente "indipendente dalla macchina" progettato dalla Digital Research Inc., abbastanza simile a quello usato dagli elaboratori con sistema operativo MS-DOS (IBM PC e compatibili).
GEMDOS fornisce accesso ad alto livello alla gestione dei file su disco (aprendo e chiudendo file, creando e aggiornando subdirectories ecc.) così come ad altre possibilità specifiche dell'hardware ST.

• *Line A*
È un insieme di funzioni grafiche fondamentali che anche il GEM usa per disegnare sullo schermo. Costituisce il "basso livello" della capacità grafica disponibile negli ST (se si esclude la memorizzazione dei bit nella memoria dello schermo).
Per quanto letto, pare che attraverso queste routine della "Line A" si potrà accedere alla gestione del "blitter chip" (bit-block transfer graphics coprocessor) e aggiornare tutto il software esistente senza bisogno di riscriverlo o ricompilarlo (direttamente dal venditore o dall'utente)
(Questo chip che è un coprocessore dedicato alla grafica, anunciato dalla Atari sin dall'estate 1986, ancora non implementato su nessun modello per qualche problema, pare, con la casa costruttrice del chip. Il suo uso permetterà logicamente di velocizzare la gestione della grafica.)

• VDI (*Virtual Device Interface*):
È una delle due parti importanti del GEM (*Graphics Environment Manager*) sempre progettata dalla Digital Research Inc.
In teoria dovrebbe fornire un set di funzioni grafiche "device-independent" in modo che gli utenti non debbano preoccuparsi del tipo di dispositivo usato, ad esempio del tipo di schermo, del tipo di stampante, anche del tipo di video-

camera. I driver di questi apparecchi sono richiamati da funzioni ad alto livello senza che ci si debba preoccupare della scrittura dei particolari.
L'idea fino a un certo punto è portata avanti sugli ST, però la sua totale implementazione (ovvero tutte le funzioni importanti su tutti i tipi di dispositivi possibili) è ancora lontana alla realtà.

• GDOS (*Graphics Device Operating System*).
Il GDOS è la parte del GEM che realizza a "basso livello" le indicazioni della VDI. È l'"alter ego" della VDI. Nella versione attuale del TOS™ negli ST i file relativi sono scomparsi dalla ROM; Atari ha annunciato ripetutamente un aggiornamento del TOS™ con i file GDOS (che ha invece il GEM in versione per macchine MS-DOS). Sviluppatori di software vicini alla direzione Atari hanno ricevuto per determinate applicazioni grafiche la versione di questi file GDOS (ad esempio Migraph che li usa in una applicazione chiamata Easy-Draw e Tom Hudson che li usa nel DEGAS Elite).
Le routine GDOS vengono implementate attraverso l'installazione di una cartella "Auto", in parole semplici vengono installate come se fossero "font", il che permette tra l'altro un aggiornamento successivo in caso di aumento del numero dei dispositivi gestibili.

• AES (*Application Environment Services*):
Questa è la seconda parte importante del GEM e consiste in una serie di routine per generare l'interfaccia utente, attraverso la chiamata a un set di funzioni come drop-down menus, finestre, finestre di dialogo ecc.
AES coordina anche il limite di funzionamento in multitasking degli accessori di scrivania (*desktop accessories*), concretamente fino a un massimo di 6.

4.2. **Pregi e limitazioni del TOS™**

Il sistema operativo di qualsiasi elaboratore non è un tutto immutabile, è "una idea in azione". Il sistema operativo, che è un insieme di programmi di utilità per gestire le risorse dell'elaboratore, viene scelto per rendere una determinata architettura hardware (CPU e interfacce) il più agevolmente manipolabile e anche programmabile.
Il TOS™ ha scelto come interfaccia utente la metafora della scrivania e l'uso dei menu con il mouse, il che comporta una gestione spinta delle possibilità grafiche dell'elaboratore.
Questa scelta, oggi generalizzata nel campo dei personal computer, partì non

molto tempo fa dall'approccio verso un'interfaccia semplice per l'utente dell'elaboratore Lisa-Apple, poi seguita del Macintosh della stessa Apple, aggiornata e ampliata dalla Commodore nel modello Amiga, e adottata già nella concezione iniziale degli ST dell'Atari.
Nel campo dei PC il sistema più usato statisticamente invece è il MS-DOS, che è quello degli elaboratori IBM PC e compatibili, e non ha un approccio simile ma un'interfaccia utente non grafica, basata su comandi immessi attraverso l'uso della tastiera alfanumerica.
Il GEM nasce in ambiente MS-DOS per fornire agli elaboratori con questo sistema operativo un'interfaccia utente con le caratteristiche sopra elencate.
Atari decide di adattare per gli ST questa concezione: MS-DOS + GEM. Scelta più o meno discutibile, il fatto è che il microprocessore MC68000 ha particolari caratteristiche diverse da quelle dei microprocessori della famiglia INTEL (i microprocessori degli elaboratori MS-DOS), il che implica una costante sensazione di incertezza negli sviluppatori di software, i quali da una parte vedono una macchina economica e graficamente potente, e anche un sistema operativo inizialmente simile a quello in cui la più parte di loro si sa muovere; le differenze di risposta e di processi reali generano però insicurezze che a volte mettono in crisi l'affidabilità della scelta.
Comunque Atari spinge il suo prodotto e il tempo comincia a darle ragione. Nel giro di questi tre ultimi anni, tra l'altro, i sistemi operativi della maggior parte delle altre macchine hanno avuto vari upgrade, cioè variazioni importanti (MS-DOS da 1.0 a 3.3, System Macintosh da 1.0 a 5.5, ecc.), invece il TOS™ aspetta ancora la sua seconda "release".
La politica Atari è stata finora quella dei "salti avanti": l'annuncio già da tempo del "blitter chip" con aggiornamento della ROM ovvero del TOS™, l'ultimissimo annuncio della scheda "transputer" abbinata ai modelli di base cosi come la concezione particolarissima della stampante laser (senza memoria interna e con comunicazione parallela con l'unità Mega). Fa pensare che il sistema TOS™ sia un sistema operativo che dovrebbe "cedere il passo" a un vero e proprio sistema diverso, probabilmente rivoluzionario, con possibilità di gestione di alto parallelismo multiprocessore e ad alta velocità (MIPS), e il tutto senza perdere compatibilità.

4.3. Altri possibili sistemi operativi per gli ST

Oltre al TOS™ esistono anche altri sistemi operativi per gli Atari ST per applicazioni multitasking come UNIX (l'OS 9 della TML e MT C-Shell della

Beckmeyer Development Tools), un emulatore Macintosh (Mac Bongo) e un emulatore MS-DOS.
I primi sono abbastanza affidabili: l'OS 9 permette di far funzionare diversi programmi scritti in C per UNIX sulle macchine ST però non permette di far funzionare i programmi TOS™. Invece MT C-Shell permette alla macchina di usare i programmi TOS™ insieme ad altri programmi specifici: come UNIX è un vero multitasking.
Gli emulatori sono disgraziatamente poco affidabili; sorprendenti sì, però è impossibile lavorarci seriamente.

4.4. **Programmare gli ST e Mega**

4.4.1. *Linguaggi di programmazione per gli Atari ST*

Spesso in passato si è discusso su quale fosse il linguaggio di programmazione da preferire; la storia dei linguaggi e le linee di tendenza per il futuro vicino lasciano pensare, però, che la torre di Babele non crollerà ancora per un bel po' di tempo.
Nel nostro caso, però, indipendentemente dalla storia generale dei linguaggi, per quanto riguarda i linguaggi applicativi sugli elaboratori Atari ST e Mega, laa risposta è semplicissima: l'attuale tendenza (più che tendenza quasi obbligo) è rivolta verso la programmazione in C.
Perché? Semplicissimo: perché il GEM è stato pensato e sviluppato in C, quindi le sue funzioni viste dal C appaiono più "naturali".
Solo in C? No, ma è meglio... Questa è la mia particolare opinione.
Però Atari ha dei vincoli con il suo passato e proiezioni verso il futuro: le macchine ST si possono programmare non solo in C ma anche con i più importanti linguaggi utilizzati fino ad oggi, e si punta anche verso qualche linguaggio più avanzato.

4.4.2. *Basic per ST*

Statisticamente il Basic è il linguaggio di programmazione più noto agli utenti degli Atari: sono disponibili più interpreti e compilatori che per qualsiasi altro linguaggio.

ST BASIC™
Produttore: Atari Corporation (USA)
Supporto: disco
Note: Dialetto Basic "interpretato" sviluppato dall'Atari, sfrutta le possibilità grafiche del GEM desktop per l'editor e offre possibilità di stampa semplici. Indicato per chi vuole cominciare con il Basic e ha un Atari ST.
Prezzo (indicativo): Gratis con l'acquisto della macchina.

Fast ST Basic
Produttore: Computer Concepts, Hertfordshire (UK)
Supporto: ROM cartridge
Note: Dialetto Basic "interpretato" molto evoluto e velocissimo per sfruttare direttamente la ROM. Eccellenti prestazioni per la gestione del GEM.
Prezzo (indicativo): 140 $

SoftWorks Basic
Produttore: SoftWorks Limited, Chicago (USA)
Supporto: Disco
Note: Dialetto Basic trasportato da altre macchine (Amiga, Commodore ecc.) molto diverso dall' ST Basic e veloce; compilatore.
Vantaggi: trasportabilità e velocità, uso ottimo se si vgliono trasportare altre applicazioni scritte in Basic e non si vogliono "vestire" con il GEM.
Prezzo (indicativo) 80 $

LDW Basic
Produttore: Logical Design Works, San Jose CA (USA)
Supporto: Disco
Note: Pacchetto che comprende Assembler, Compiler e Linker, sfrutta la stessa versione dialettale dell'ST Basic. È l'unico compilatore che può compilare direttamente programmi scritti in ST Basic.
Prezzo (indicativo) 70 $

MemSoft Basic
Produttore: MemSoft, Nizza (Francia)
Supporto: Disco
Note: Basic interpretato molto apprezzato dagli utenti tradizionali Basic, favorisce la portabilità da altre macchine.
Vantaggi: Trasportabilità e supporto tecnico.
Prezzo (indicativo): Gratis con l'acquisto della documentazione ≈ 25 $

GFA Basic
Produttore: MichTron, Michigan (USA)
Supporto: Disco editor + disco compilatore
Note: Basic molto potente, gestione semplicissima del GEM, diverso del dialetto ST Basic, molto ben accetto nel mondo Basic.
Vantaggi: Trasportabilità e velocità, ottimo se si vogliono trasportare altre applicazioni scritte in Basic per "vestirle" con il GEM.
Prezzo (indicativo): 80 $ editor +80 $, compilatore ≈ 160 $

Philon Fast/Basic-M
Produttore: Philon Inc., New York (USA)
Supporto: Disco editor + disco compilatore
Note: Totalmente compatibile con il Basic Microsoft, permette di compilare per ST programmi scritti per altre macchine con Basic Microsoft (che è uno dei Basic più diffusi)
Prezzo (indicativo): 130 $; importato in italia ≈ L. 249.000

Omikron Basic
Produttore: Omikron Software, Aachen (Germania occ.)
Supporto: ROM cartridge e disco
Note: Basic compilato compatibile Microsoft con possibilità grafiche evolute, utilizzo del mouse, alta velocità. Interessante la libreria di gestione della porta MIDI.

4.4.3. *Pascal*

Il Pascal è il linguaggio più conosciuto a livello scolastico e accademico, è molto ben strutturato e permette di realizzare qualsiasi applicazione; la qualità del software prodotto dal Pascal è alta, grazie alla rigidità della sua concezione strutturale.

MCC Pascal
Produttore: Metacomco, Scotts Valley CA (USA)
Supporto: Disco con Editor, Compiler e Linker
Note: ISO 7187 standard Pascal, usato già per l'Amiga e per il Sinclair QL, viene abbinato all'MCC Macro Assembler. La documentazione è troppo concisa, esiste un'utility per la generazione simplificata di menu (MENU+).
Vantaggi: Metacomco è un'azienda che commercializza anche il Lattice C, il

Cambridge LISP, il BCPL e il Macro Assembler; quest'ultimo è comune a C e Pascal il che rende l'idea dell'assoluta parità di prestazioni del Pascal e del C in questo ambiente. Metacomco fornisce anche librerie di utility aggiornate.
Prezzo (indicativo): 100 $

UCSD Pascal
Produttore: Pecan Software Systems, New York (USA)
Supporto: Disco con Editor, Compiler e Linker
Note: È un dialetto del Pascal di Niklaus Wirth, modificato per semplificarne l'uso; ha una documentazione generale di 600 pagine +100 pagine specifiche dedicate agli ST; combina il compilatore con un gruppo di "developement tools" chiamati p-System.
Vantaggi: È un Pascal più semplice da usare rispetto al Pascal standard.
Prezzo (indicativo): 100 $

Pro Pascal
Produttore: Prospero Software Lmtd., Londra (UK)
Supporto: Disco con Editor, Compiler e Linker
Note: Versione completa dello ANSI 770 × 3.97 standard Pascal compiler, ha 7 e 16 cifre di precisione per l'aritmetica in virgola mobile, compilazione separata e 4 byte integers. Completamente integrate le funzioni GEM AES e VDI.
Ventaggi: Trasportabilità di programmi scritti per altre macchine agli ST.
Prezzo (indicativo): 130 $

Personal Pascal
Produttore: Optimized Systems Software Inc., San Jose CA (USA)
Supporto: Disco con Editor, Compiler e Linker
Note: Senza dubbio il sistema di sviluppo che ha avuto le più ferventi adesioni nel campo dei critici americani e anche italiani.
Il Compiler supporta l'ISO Pascal con estensioni per stringhe, compatibile con Turbo Pascal, file system e GEM; ammette compilazione modulare e riferimenti a procedure e funzioni esterne, interagisce con l'editor a cui può ritornare a seguito del rilevamento di un errore, e con il linker, che può lanciare a compilazione ultimata. Documentazione soddifacente.
Prezzo (indicativo): 80 $; importato in Italia ≈ L. 120.000

4.4.4. *Modula - 2*

È un linguaggio relativamente nuovo, disegnato da Niklaus Wirth (il creatore del Pascal), di lessico ibrido tra C e Pascal.
Rispetto al Pascal offre supporti importanti per la programmazione modulare, per la concorrenza (una strada a cui gli Atari ST puntano) per il multitasking tra diversi programmi, offre gestione ad alto livello per le operazioni a basso livello della macchina e librerie standardizzate per il trasporto di programmi tra diversi sistemi operativi.

Modula-2 /ST
Produttore: TDI Software Inc., Dallas Texas (USA)
Supporto: Tre versioni diverse:
- *a*) Disco con Editor Compiler e Linker
- *b*) Disco con Editor Compiler Linker + disco delle librerie standard
- *c*) Disco con Editor Compiler e Linker + disco librerie standard + disco Debugger

Note: Questo TDI Modula-2/ST ha le librerie del GEM e accede alla gestione della *Line A*, si presenta con tre versioni, una per inizializzare la programmazione in Modula-2, un altro per quelli che vogliono sviluppare software e un terzo indirizzato alla produttività del software.
È un linguaggio nuovo che permette di "linkare" programmi di C e Pascal (in *TDI* e *C* è Megamax)
Prezzo (indicativo) (*a*) 80 $; (*b*) 150 $; (*c*) 230 $; solo Debugger 70 $

Modula-2 Compiler / Toolkit
Produttore: Modula 2, Bristol (UK)
Supporto: Disco e manualistica
Note: Questo Modula-2 ha un'implementazione completa, con debugger, resource construction set, diverse librerie del GEM e accede alla gestione della Line A. Sono forniti anche demo e programmi di utilità. È la versione originale (svizzera) di Niklaus Wirth.
Prezzo (indicativo): ≈ L. 149.000

4.4.5. *LOGO*

LOGO è un linguaggio sviluppato al MIT (Massachusetts Institute of Technology) da Seymour Papert e la sua equipe.
La sua origine risale alla fine degli anni sessanta. Oggi è il linguaggio più noto per avvicinare i bambini all'uso degli elaboratori. Sono due i motivi per i cui il linguaggio si è diffuso solo negli ultimi anni:

- si stava completando e sperimentando al MIT e non c'era alcuna fretta di commercializzazione,
- richiede macchine con capacità di memoria superiore e grafica, che fino a poco tempo fa erano fuori dalla portata dei centri scolastici; lo sviluppo tecnologico ha reso possibili macchine economiche con quelle caratteristiche solo all'inizio degli anni ottanta.

Gli ST offrono un ambiente idoneo per questo tipo di approccio ed era normale che una versione di LOGO prima o poi venisse sviluppata per queste macchine. Atari non ha voluto però lasciarla all'improvvisazione e ha sviluppato direttamente un dialetto del LOGO per le sue macchine, che offre come servizio a richiesta (come fa anche con l'ST Basic).

La geometria della tartaruga

Come può avvenire che, grazie al LOGO, anche bambini in età prescolare entrino in rapporto attivo con l'elaboratore, programmando una macchina che appare così difficile da usare?
Il fatto che il LOGO sia semplice non basta; occorre anche che i bambini abbiano una motivazione, uno stimolo, un interesse concreto a usarlo.
Tutto questo è reso possibile da un sottoinsieme del LOGO: *la geometria della tartaruga* (*Turtle graphics*).
La tartaruga era in origine un robot da pavimento, in grado di muoversi lasciando dietro di sé una traccia colorata. Successivamente, per evidenti motivi di economicità, è stata fornita anche una simulazione della tartaruga sullo schermo dell'elaboratore.
Il movimento della tartaruga è controllato dal bambino attraverso alcuni semplici comandi; con questi semplici comandi inmediatamente riferibili anche al movimenti del proprio corpo, e quindi facili da capire, il bambino può realizzare disegni anche complessi senza doversi sottoporre a un lungo e

noioso periodo di addestramento per impratichirsi del linguaggio. Inizialmente non deve neppure scrivere un programma vero e proprio. Grazie alla interattività di LOGO, può infatti far eseguire un'istruzione alla volta, seguendo sullo schermo lo svilupparsi della propria creazione, e decidendo di volta in volta l'istruzione successiva. Solo in seguito, se ne sentirà l'esigenza, imparerà a raccogliere in una "procedura" (o programma) le istruzioni relative a un certo disegno, a memorizzarle, e così via.

ST LOGO™
Produttore: Atari Corporation (USA)
Supporto: Disco
Note: Versione ampliata del LOGO, adattata alle macchine ST; sviluppata dall'Atari sfrutta le possibilità grafiche del GEM desktop, il mouse e i menu. Indicato per capire cos'è la programmazione per bambini da 3 a 90 anni.
Prezzo (indicativo): A richiesta, con spese di documentazione e spedizione a carico dell'utente.

4.4.6. *C language*

Ognuno dei linguaggi precedenti (Basic, Pascal-Modula 2, LOGO) ha una presenza storica in ambiente informatico che giustifica totalmente la necessità di interpreti, compilatori e kit di sviluppo per gli ST.
Per il C la realtà storica si abbina alle scelte fatte da Atari e Digital Research Inc. nel campo del sistema operativo (TOS™-GEM).
Non è il caso di dilungarsi sul motivo per cui il C sta diventando lo standard nello sviluppo di programmi applicativi nell'area dei personal (non solo Atari, ma anche mondo MS-DOS, Commodore Amiga e Macintosh Apple), ma val la pena di evidenziare qualche aspetto significativo.
Il C è un linguaggio strutturato che permette più libertà che il Pascal: permette di lavorare in modo meno "pre-organizzato" o anche "più anarchicamente". È abbastanza indipendente dalla macchina, il che permette la portabilità di programmi tra diversi elaboratori. Permette anche lo sviluppo "a moduli" a se stanti, il che favorisce il lavoro in equipe. Le sue librerie di programmi di utilità permettono di usufruire di un significativo supporto alla programmazione.
Il C, avendo già dietro di sé una grande cultura informatica, si trova implementato su macchine con sistemi operativi che vanno dall'UNIX al CP/M68K passando per l'MS-DOS.
Questo agli sviluppatori di software è risultato di grande comodità, e ha

contribuito in buona misura a fare del C lo standard per applicazioni "creative" in area "personal" e anche "mini".
Atari consiglia l'uso di un KIT di sviluppo per programmatori basato sull'uso del C abbinato a un Assembler (vedere Development KIT).
Però l'offerta in campo C non è solo quella: vediamo tutti i C preparati per gli ST.

Megamax C
Produttore: Megamax Inc., Richardson Texas (USA)
Supporto: Disco con Editor, Compiler Linker e librerie
Note: Completa l'implementazione del C di Kernigan & Ritchie; supporta floating point, overlays, ricorsione, processi e librerie batch, ha un costruttore di risorse per creare oggetti GEM. Documentazione completa delle routine GEM, AES e VDI
Prezzo (indicativo): 200 $

Mark Williams C
Produttore: Mark Williams Company., Chicago (USA)
Supporto: Tre diverse possibilità:
a) 2 dischi con Editor, Compiler, Linker e librerie
b) *Let's C*, disco con Editor e comandi riassunti
c) CSD debugger
Note: Implementazione completa del C di Kernigan & Ritchie con le estensioni più recenti implementate sotto UNIX. Il prodotto include un assembler lettore e archiviatore, un debugger simbolico, il C-Shell, utilities, il Micro EMACS e un editor "a tutto schermo".
Il *Let's C* contiene un sottoinsieme del Mark Williams C con il codice più denso con il Micro EMACS e l'editor.
Il CSD è un debugger del codice sorgente che permette di lavorare indipendentemente dalla macchina.
Vantaggi: Questa implementazione è totalmente compatibile con il Mark Williams C per ambiente MS-DOS, il che permette di trasportare semplicemente applicazioni scritte in ST a IBM PC e compatibili e viceversa.
Prezzo (indicativo): (*a*) 180 $; (*b*) 75 $; (*c*) 75 US $; importato in Italia ≈ L. 270.00. (solo *a*)

Lattice C
Produttore: Metacomco., Scotts Valley CA (USA)
Supporto: Due diverse possibilità
a) 2 dischi con Editor, Compiler, Linker e MENU +, librerie
b) Disco con solo MENU +
Note: Implementazione completa del C di Kernigan & Ritchie. Il Lattice C per MS-DOS è uno dei compilatori più conosciuti (con questo compilatore sono state scritte tutte le routine del GEM), e permette di trasportare facilmente applicazioni scritte in ST a IBM PC e compatibili e viceversa. Si può acquistare con MENU + che è un insieme di "utility" di sviluppo software (riunisce le funzioni Batch e Shell in unico programma); è anche abbinabile al Micro Assembler della stessa ditta.
Vantaggi: La Metacomco commercializza non solo questo Lattice C per gli ST, ma anche il MCC Pascal, il Cambridge LISP, il BCPL e il Macro Assembler; fornisce anche librerie di utility aggiornate.
Prezzo (indicativo): (a) 150 $; (b) 30 $ (solo MENU+); importato in Italia ≈ L.199.000. (tutto) o L. 65.000 (solo MENU+)

GST- C
Produttore: GST Holdings Lmtd., Cambridge-(UK)
Supporto: Due diverse possibilità:
a) 2 dischi con Editor, Compiler, Assembler, Linker e librerie
b) Disco con un optional chiamato "Fractal Factory, source code"
Note: Il prodotto è composto da compilatore (che non supporta floating point), Linker (GST-Link), Assembler (GST-ASM) e uno stupendo Editor (GST-Edit).
Prezzo (indicativo): (*a*) 80 $; (*b*) 20 $

DEVELOPMENT KIT Atari ST
Produttore: Atari Corporation (USA)
Supporto: 5 dischetti e manualistica (più di 1000 pagine)
Note: Il prodotto è composto da compilatore C (che usa la particolare versione Alcyon C - 68000), Linker, Assembler, Editor, Debugger, programmi di utility e tutta la manualistica completa di sistema (VDI AES GDOS ecc.). Solo in inglese.
Prezzo (indicativo): ≈ L. 490.000

4.4.7. *FORTRAN*

Il FORTRAN è stato il primo dei linguaggi strutturati ad alto livello: la prima versione fu sviluppata in casa IBM nel 1956. Il nome significa *FORmula TRANslator* ed era molto appropriato per il calcolo numerico. Ha avuto successivi aggiornamenti, e oggi il FORTRAN più diffuso è l'ANSI Fortran 77. Il fatto che sia ancora molto utilizzato è dovuto alla grande quantità di cultura informatica che si porta dietro e soprattutto alla grande qualità dei suoi compilatori.

AC/ Fortran
Produttore: Absoft, Royal Oak USA)
Supporto: Disco e manualistica
Note: Linguaggio ANSI Fortran 77. Il prodotto è composto da compilatore che accetta l'aritmetica in virgola mobile, Linker, gestore di libreria e un source level debugger "a tutto schermo". Il programma compilatore occupa solo 48K in RAM, è scritto in Assembler per velocizzare la compilazione e usa GEM come interfaccia.
Prezzo (indicativo): 200 $

PRO- Fortran 77
Produttore: Prospero Software Lmtd, Londra (UK)
Supporto: Disco e manualistica
Note: Implementazione completa dello standard ANSI Fortran 77. Particolarità interessante, il programma controlla la corretteza della sintassi dei comandi scritti, il che permette la traduzione immediata del codice scritto in ambiente mainframe.
Prezzo (indicativo): 150 $

4.4.8. *Forth*

È un linguaggio di origine relativamente recente (inizi degli anni settanta), disegnato da Charles Moore per applicazioni in astronomia. Successivamente il linguaggio è stato utilizzato per altri tipi di applicazioni, soppratutto nel campo degli "home computer" a 8 bit; il vantaggio è che inizialmente è semplice da usare, ma allo stesso tempo è molto conciso, e produce codice molto compatto, il che consente un risparmio enorme di memoria.
I "tifosi forth" che si raggruppano in una associazione, la FIG (Forth Interest

Group) si divertono a confrontarlo con il COBOL (Davide contro Golia): la metafora è bella, home computers contro mainframe, però non rappresenta la realtà dei problemi in informatica; Davide e Golia, Forth e COBOL, sono in una guerra che a noi interessa ben poco. Forth non serve a risolvere problemi di concorrenza, di alto parallelismo; è adatto alla descrizione/simulazione di processi sequenziali ma questi sono solo una piccola parte dei processi reali, d'altra parte è molto criptico e ha alla base alcune scelte (memory stack, assegnazioni FILO ecc.) che limitano moltissimo la sua versatilità.
Si può dire che è un linguaggio particolarmente adatto a macchine con poche possibilità di memoria; per fortuna oggi almeno a questo livello si può lavorare con più comodità.
Comunque i suoi difensori in ambiente Atari non sono pochi, soprattutto negli USA, e pertanto si trovano sul mercato diversi compilatori e kit di sviluppo per questo linguaggio adatti alle macchine ST.

4 ×Forth
Produttore: The Dragon Group, Elkview (USA)
Supporto: Disco e manualistica
Note: Implementazione completa del Forth standard '83.
Prezzo (indicativo): 100 $

Multi-Forth
Produttore: Creative Solutions Inc., Rockville (USA)
Supporto: Disco e manualistica
Note: Implementazione completa del Forth standard '83, Mac compatibile.

Mach 2
Produttore: Palo Alto Shipping, Menlo Park CA (USA)
Supporto: Disco e manualistica
Note: Sistema di sviluppo multitasking Forth standard '83 e compilatore basato sulla implementazione per Macintosh, estremamente adatto all'ambiente GEM. È abbinato a un Motorola Assembler e a un symbolic debugger.
Prezzo (indicativo): 100 $

4.4.9. *Assembler e Macro Assembler*

Programmare in Assembler è difficile ma bello, se avete voglia e tempo. L'Assembler è il linguaggio più vicino al linguaggio macchina e offre la possibilità di gestire "quasi direttamente" l'hardware.
In realtà tutti i linguaggi di alto livello (tutti quelli di cui abbiamo parlato in precedenza lo sono) hanno bisogno di essere trasformati in Assembler per ottimizzarsi e poi per effettuare la definitiva conversione del codice in linguaggio macchina.
La programmazione in Assembler è assolutamente necessaria quando si vogliono generare applicazioni "eccezionali" non presenti nei linguaggi di alto livello.
In informatica musicale un'ottimizzazione dei processi di elaborazione passa neccessariamente per la scrittura in Assembler di determinate procedure a basso livello per scopi normalmente non previsti dai costruttori. Cosi si possono ottimizzare i processi di elaborazione e sintesi numerica del suono e si possono costruire i "driver" dell'interfaccia MIDI per quei calcolatori che non li hanno.
I linguaggi Assembler logicamente non sono affatto "indipendenti dalla macchina" però ci sono famiglie di Assembler CPU-dipendenti e il Motorola 68000 è una CPU assai utilizzata da diversi costruttori per le varie macchine (da Apple per i suoi Lisa, Macintosh e Apple II GS, da Commodore per l'Amiga, da Sinclair per i QL, da Atari per gli ST ecc.) e di conseguenza ha una buona quantità di Assembler di alta qualità realizzati da diversi costruttori.
Quello che si richiede agli Assembler è la possibilità di dialogo sia con la particolare architettura hardware della macchina sia con compilatori di linguaggi ad alto livello. È per questo che gli Assembler vengono caratterizzati da questi due estremi: ad esempio Assembler per la macchina con architettura "A" capace di dialogare con il compilatore "B". Normalmente gli Assembler si trovano abbinati ai compilatori nei kit di sviluppo.
Quando un Assembler deve parlare con più compilatori, sono necessarie funzioni di traduzione automatica per ciascuno dei particolari compilatori: queste funzioni sono denominate "macro" e questi prodotti vengono chiamati Macro Assembler.
Questi prodotti sono già stati citati parlando dei compilatori.

4.5. Intelligenza Artificiale e ST

L'Intelligenza Artificiale è già una realtà! (Il problema reale è capire che cosa vuol dire esattamente Intelligenza Artificiale.)
L'Intelligenza Artificiale comprende una serie di linguaggi adatti a modi diversi di intendere la programmazione (programmazione non procedurale, funzionale o logica), una concezione dell'uso di determinati hardware (Lisp Machine, calcolatori della quinta generazione, reti di automazione d'ufficio, *flexible manufacturing* e robot), una seria riflessione sui meccanismi della conoscenza umana, qualche applicazione avanzata (decision maker, sistemi esperti) e anche una enorme quentità di fumo.
Probabilmente la popolarità dell'Intelligenza Artificiale è dovuta almeno in parte alla musicalità e alla magia del nome... Per gli Atari ST, comunque, esiste qualche approccio serio ai temi che vanno sotto il nome di IA o AI (nella terminologia anglosassone). Vediamo quali linguaggi siano disponibili per questo settore.

4.5.1. *Lisp*

Contrariamente a quello che si potrebbe pensare, il Lisp ha un'anzianità di servizio pari a quella del più vecchio fra i linguaggi procedurali strutturati e sequenziali: è coetaneo infatti del FORTRAN, essendo stato progettato da John McCarthy proprio nel 1958.
Il nome sta per *LISt Processing* (elaborazione di liste); è stato il primo linguaggio non procedurale, funzionale. Lo stile di programmazione e la struttura dei programmi in Lisp si basano non sull'architettura della macchina, ma sulla matematica: un programma Lisp non è una sequenza di istruzioni elementari di trasferimento di dati dalla memoria all'unità centrale e viceversa o di operazioni su di essi, bensì un insieme di funzioni matematiche composte tra loro, in modo che ognuna di esse riceva come dati i risultati della precedente.
Il Lisp non possiede uno standard universalmente accettato perche dal lontano 1958 è stato usato a livello di ricerca ed ha subito delle variazioni d'"autore", il che lo ha reso fino a un certo punto inefficace.
L'altro motivo della sua scarsa efficienza è che il Lisp non è adatto alla architettura Von Neumann (l'architettura hardware standard degli elaboratori fino ad oggi) come lo sono invece i linguaggi procedurali o imperativi. Caso unico finora nella storia dell'informatica, di fronte all'impossibilità di adat-

tarlo alle macchine standard, ha stimolato la costruzione di architetture hardware specifiche, le “Lisp Machine”, che permettono di eseguire programmi in Lisp con la stessa efficienza con cui i calcolatori con architettura Von Neumann eseguono programmi in FORTRAN o in C.
Queste le caratteristiche del Lisp come linguaggio di programmazione:

- Linguaggio funzionale, in cui il meccanismo fondamentale per la costruzione di funzioni complesse è la composizione di funzioni più semplici;
- Progettato per elaborare espressioni simboliche piuttosto che numeri;
- Rappresenta sia i dati che i programmi nella stessa forma, per cui non è difficile scrivere programmi che leggano, modifichino, costruiscano altri programmi in modo semplice;
- Dispone di una base matematica solida, la teoria del “lambda calcolo”, che consente di stabilire in modo rigoroso quale è la funzione definita da un programma Lisp, e quindi di verificare la correttezza dei programmi;
- Consente di scrivere programmi ricorsivi, che cioè richiamano se stessi al proprio interno;
- È altamente interattivo.

Queste caratteristiche hanno favorito lo sviluppo di tecniche di programmazione avanzate (programmazione funzionale e ricorsiva, programmazione guidata dai dati, programmazione in ambiente interattivo, verifica della correttezza dei programmi) di enorme importanza. Inoltre, le sue capacità di elaborazione simbolica lo hanno reso il linguaggio d’elezione nelle applicazioni in campi come la robotica, la comprensione del linguaggio naturale, la traduzione automatica, la risoluzione automatica di problemi.

Cambridge LISP
Produttore : Metacomco, Scotts Valley CA-(USA)
Supporto: Disco con Editor, Compiler e Linker
Note: Usato già per l’Amiga fornisce un ambiente interessante per iniziare la conoscenza del Lisp. Include la gestione delle librerie grafiche VDI, AES e GEM e si può allacciare a MENU+ ed altre utilities MCC.
Prezzo (indicativo): 200 $; importato in Italia ≈ L. 340.000

4.5.2. *Prolog*

Il Prolog, svilupppato in Francia all'Università di Marsiglia nel 1972 da Alain Colmerauer, è fondamentalmente diverso dal Lisp: il suo nome sta per *PROgramming in LOGic.*
Tanto la programmazione logica quanto la programmazione funzionale (quella del Lisp) rientrano nell'ambito della programmazione non procedurale. Come è successo per il Lisp, il Prolog è rimasto a lungo confinato a livello di ricerca sperimentale, ma il lancio, nell'ottobre del 1981, del progetto di costruzione degli elaboratori della quinta generazione, che punta in modo esplicito sulle tecniche della programmazione logica, ed è sostenuto da colossali finanziamenti da parte del governo giapponese, ha improvisamente fatto divampare l'interesse intorno al Prolog (che è stato ufficialmente scelto dai giapponesi come linguaggio di base per queste future macchine).
Oggi, questo linguaggio è ampiamente usato nell'area dell'AI, in particolare per la produzione di "sistemi esperti", cioè programmi molto flessibili, in grado di fornire "consulenze", esattamente come un esperto umano, in settori professionali (si possono ad esempio citare la diagnosi medica, la progettazione ingegneristica, la consulenza per investimenti finanziari, ecc.).

Che cos'è la programmazione logica?
L'idea di base della programmzione logica è che un programma deve essere costruito *non specificando le operazioni* da eseguire per risolvere il problema, come nella programazione procedurale, *ma descrivendo il contesto* in cui si enuncia il problema.
Tale contesto è costituito da "oggetti" tra cui valgono delle "relazioni". La descrizione del contesto consiste nell'elencare l'insieme delle relazioni tra gli oggetti del contesto che si sa essere vere.
Un programma in Prolog è costituito da due parti:

- La dichiarazione di un insieme di "fatti" (cioè di proposizioni vere) che riguardano gli oggetti e le loro relazioni;
- La definizione di nuove relazioni o di regole di trasformazione delle relazioni.

L'insieme dei fatti e delle definizioni di relazioni costituisce la *base di conoscenza* su cui possiamo operare. (Il livello di libertà e creatività nella programmazione logica nasce precisamente dalla generalità dei concetti

"oggetto" e "relazioni".)
L'esecuzione di un programma Prolog è attivata da una "domanda" posta dall'utente intorno agli oggetti ed alle loro relazioni. Il programma esamina i fatti inclusi nella base di conoscenza, trae le conclusioni del caso, e risponde all'utente; inoltre, può anche inserire la risposta come nuovo fatto nella base di conoscenza, arricchendola.

Personal Prolog
Produttore: Optimized Systems Software Inc., San Jose CA (USA)
Supporto: Disco e manuali
Note: Prodotto dalla stessa ditta che ha realizzato il Personal Pascal.
Prezzo (indicativo): 80 $

4.5.3. *Coda*

Questa lista abbastanza esauriente presenta due delle tre possibilità di sviluppo con linguaggi di AI; la terza possibilità invece non è ancora presente in ambito ST ed è rappresentata dai linguaggi *object-oriented* (ovvero orientati agli oggetti) come lo SmallTalk™ sviluppato dalla Xerox™.

4.6. **Qualche idea su come sviluppare software per gli ST e qualche utility sulla MIDI**

La particolare costruzione del TOS™ degli Atari ST permette di sviluppare software a due livelli principalmente, e ad un terzo in casi eccezionali:

- Senza la grafica;
- Sfruttando la grafica standard ovvero le routine VDI e AES;
- Sfruttando tutte le possibilità della macchina attraverso la *Line A*.

Diversi dei compilatori sopra elencati permettono i due primi approcci. Il primo si definisce "programmare in TOS™", il secondo "applicazione in ambiente GEM".
Questo è possibile perche gli ST hanno un sistema "veramente" modulare.
Il primo modo di operare, ovvero scrivere sotto TOS ™ e compilare in seguito, è il modo più veloce per testare l'algoritmo del programma (nel caso in cui l'applicazione sia più concettuale che grafica).

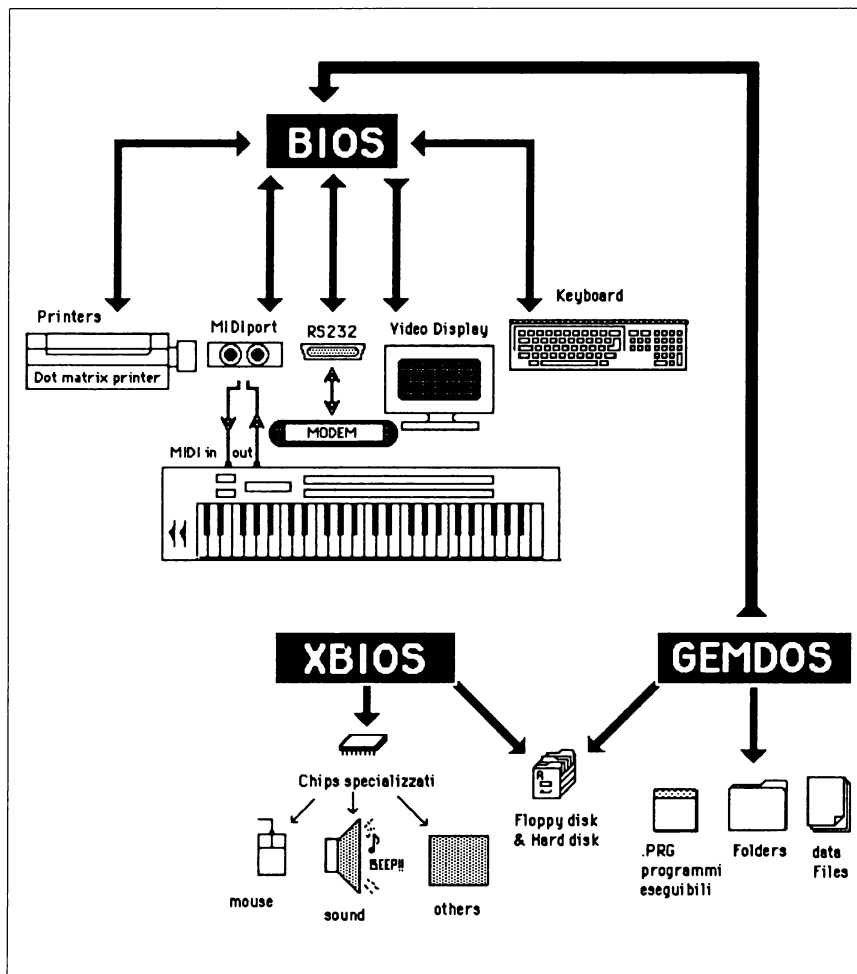



Fig. 4.1 Il TOS™ (The Operating System) dell'Atari e la sua struttura modulare.

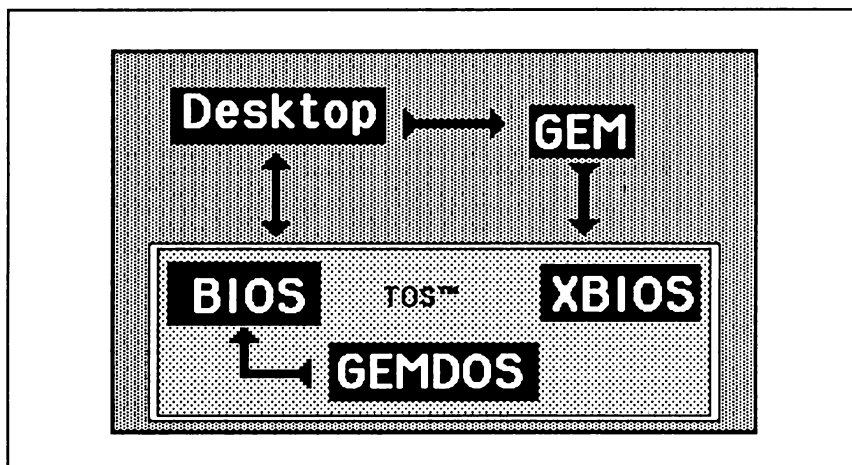



Fig. 4.2 La ROM degli Atari ST™ e Mega.

Dopo si può utilizzare il GEM per "vestire" l'applicazione e compilarla vedendo i risultati riproporsi il tutto in modo più interattivo, e cosi via.
Impadronirsi delle routine fondamentali del GEM non è cosa difficile, ma dipende abbastanza dal tipo di linguaggio utilizzato. Lavorare in C (già l'abbiamo detto) è più semplice ed efficace, però può darsi che sia più difficile imparare il C che continuare a usare il linguaggio al quale si è già abituati.
Il terzo modo di programmare richiede necessariamente che si lavori in Assembler: è questo il modo in cui si realizzano in genere i videogiochi con grafica spinta e veloce, cosi come le applicazioni avanzate di elaborazione numerica del suono.

4.6.1. *Scrivere software MIDI per un ST*

Il TOS fornisce diverse funzioni per la comunicazione di dati MIDI tra una periferica MIDI e l'elaboratore. Questa è una differenza importante, rispetto agli altri elaboratori, per i quali ci si deve arrangiare nella scrittura delle procedure (tra l'altro molto critiche a livello di "timing" e necessariamente in Assembler) per gestire attraverso la RS232 l'interfaccia MIDI ovvero la UART.
Gli ST ti danno il lavoro già fatto e accessibile da linguaggi di alto livello (Pascal, C, Basic, ecc); vediamo come si può fare dal C (per altri linguaggi

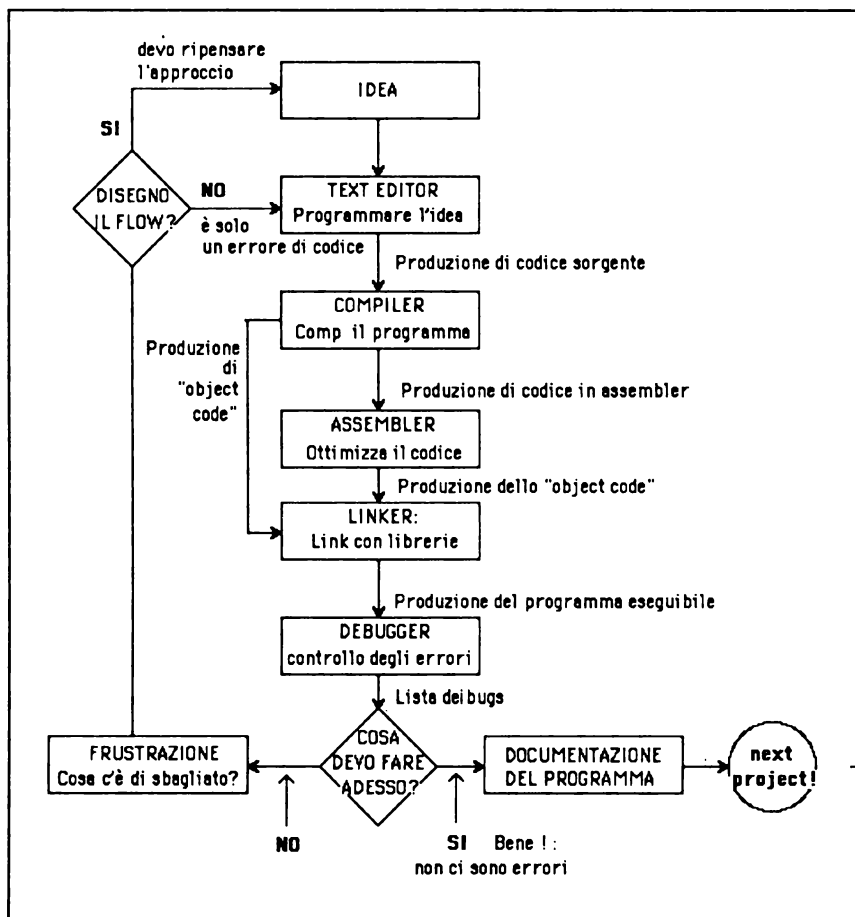



Fig. 4.3 Il processo comune di produzione del software a livello semi-professionale (il più normale per gli utenti di Atari ST).

consulta il manuale del compilatore). Nel caso volessi farlo in Assembler (per certe applicazioni spinte, come abbiamo già detto, sarebbe meglio) devi prendere questi comandi delle librerie C e dissasemblarli.
Per inviare informazioni dalla porta MIDI-out dello ST, prepara un array o una stringa di byte con le informazioni in un formato simile a questo:

```
static char midiinfo[] = {scrivi qui i dati MIDI};
/*80 byte al massimo per ogni chiamata alla Midiws*/
```

Fai i conti del numero di byte che vuoi spedire, e poi spediscili in un formato simile a questo:

```
Midiws(count-1,midiinfo);
```

Invece, se vuoi prendere informazioni in byte dalla porta MIDI in, devi usare le funzioni del BIOS `Bconstant()` e `Bconin()` in un formato simile a questo:

```
unsigned char midibyte;
if(Bconstant(3) ==-1)
/*3 indica al BIOS di guardare verso la MIDI*/
midibyte = (unsigned char)Bconin(3)
```

Atenzione! Alcuni dei compilatori C fanno cose strane con i caratteri di codice maggiore di 127. Questi compilatori non accettano di buon grado un numero di otto bit "unsigned", cioè senza segno, e possono fare scherzi cambiando il segno; ad esempio, 128 può diventare –128, o qualche altro valore.

4.7. **Una nota finale**

Un ultimo consiglio: non usate programmi compilatori e kit di sviluppo copiati e senza documentazione! Se non li trovate in Italia, i compilatori e la relativa documentazione sono sempre accessibili, basta scrivere ai fornitori originali (tutte le indicazioni del caso si trovano nelle appendici di questo libro).
E se è per il prezzo, a copiare non risparmiate nulla, al contrario: perché il vostro tempo è davvero oro.

Tra l'altro, grazie per averlo speso leggendo fin qui!

Appendice 1

Software musicale commercializzato (ottobre '87) per gli Atari ST e Mega

C'è una enorme offerta per gli Atari ST a livello di software musicale e pseudo musicale, ma disgraziatamente l'enorme quantità di programmi disponibili non significa invece che esistano tutti i programmi per tutte le applicazioni significative o necessarie né che i programmi esistenti offrano una grande qualità. In effetti, molti di questi programmi non raggiungono il livello di qualità minimo per lavorare. L'approccio degli sviluppatori di software di alcuni programmi non è neanche amatoriale.
La musica è un fenomeno omnipresente in tutte le culture però allo stesso tempo è una enorme sconosciuta che molti pretendono di presentare in modo troppo banale. Per fare musica è necessario software di qualità: basterebbe un solo programma per ogni applicazione tipo, però di qualità accettabile, flessibile e con possibilità di ulteriori perfezionamenti (*upgrade*) senza perdere compatibilità (ovvero senza perdere tutto il lavoro fatto sulle versioni precedenti).
Per questo motivo all'inizio di ogni elenco di "software tipo" esaminerò più in dettaglio uno o più programmi che mi sembrano degni di evidenza, per il livello

di serietà e la qualità dell'impostazione. In seguito elencherò tutti i software della stessa categoria già commercializzati, in ordine alfabetico e in modo succinto, avvertendo delle mancanze più significative (almeno entro i limiti di ciò che sono riuscito ad appurare).

I programmi che accettano la scrittura notazionale

I programmi che accettano la scrittura notazionale sono conosciuti come *Composer* e possono essere abbinati o meno a sequencer, in modo da trasformare la scrittura in una sequenza di eventi MIDI e viceversa.
Vediamo due approci interessanti: il primo è quello tradizionale del "composer", il secondo è un ibrido intelligente, didattico e innovativo, tra sequencer e composer:

The Copyst™
Produttore: DR T'S Music Software (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Uno dei programmi per la scrittura notazionale più conosciuto in ambiente MS-DOS e ri-compilato per gli ST.
Ha a disposizione la maggior parte dei simboli musicali (tutti i fondamentali, mancano solo i sessantaquattresimi) con limitazione a un numero di note proporzionale alla memoria disponibile (ok). Accetta tutte le stampanti standard DMP e la Laser-jet HP (non quelle Postscript), ha la possibilità di generare "font personalizzate" (segni particolari tanto in DMP come in laser), ha un text editor incluso per i testi, trasposizione automatica, stampa il totale o le "parti" interattivamente, il mouse invece è "optional" il che lo rende un po' macchinoso da usare inizialmente.
Compatibilità: Legge e scrive su i sequencer KCS e MRS della DR T's.
Hardware speciale: No
Valutazione complessiva: Buono. Livello di stampa non sufficiente per l'editoria, ma valido per arrangiamenti, società di autori ecc.
Prezzo (indicativo): Importato in Italia ≈ L. 270.000

Il Compositore SIAE™
Produttore: Marco Fossen (Italia)
Distribuzione italiana: Atari Italia (Milano)
Note: Pacchetto sequencer + composer integrati e interattivi. Il sequencer ha 16 tracce, il composer 4 righi musicali, ma la stampa ha i 16 righi (che

corrispondono alle tracce del sequencer). Input da tastiera musicale MIDI, tastiera alfanumerica o mouse, tanto in tempo reale come step by step, visualizzazione in tempo reale di ciò che si suona. Edit inmediato con funzioni "mouse" asegnabili. Riascolto tanto dalla periferica MIDI come dal "chip" di suono interno dell'Atari.
Sistema di composizione per Icona che rappresentando la tastiera di guitarra la tastiera di pianoforte ed anche una batteria. Stampa standard accetabile per le società di autori tramite stampanti DMP.
Compatibilità: Con le succesive versioni
Valutazione complessiva: Bellisimo. Ideato per gli ST e per essere utilizzato da utenti con necessità di traduzione inmediata a partitura dei propri lavori, senza bisogno di utilizzare un "composer" troppo macchinoso, è un programma ottimo perché sfrutta al massimo le possibilità grafiche ed evocative della macchina, a un prezzo più che contenuto.
Prezzo (indicativo): ≈ L. 100.000

Altri:

DIATONICA-7™
Produttore: Quiet Lion (USA)
Distribuzione italiana: no
Note: Programma per la composizione assistita dall'elaboratore, permette la armonizzazione (non in modo stretto), ed è un aiuto alla orchestrazione.
Compatibilità: Con gli altri prodotti della Quiet

EZ - Score ST™
Produttore: Hybrid Arts (USA)
Distribuzione ialiana: Mack Music (Roma)
Note: Stampa di partiture generate dai sequencer della Hybrid.
Compatibilità: EZ track, Sync Track e SMPTE Track

K Mistrel™
Produttore: Kuma (UK)
Distribuzione italiana: Atari Italia (Milano)
Note: Programma molto elementare, con uso del mouse e della MIDI.
Prezzo (indicativo): Importato in Italia ≈ L. 50.000

MUSIC BOX™
Produttore: XLEnt Software (USA)

Distribuzione italiana: no
Note: Otto tracce, solo step input

Master Score ™
Produttore: Steinberg
Distribuzione italiana: Midi Ware
Note: Kit per la stampa di partitura.
Compatibilità: con gli altri prodotti della Steinberg.

Music Studio™
Produttore: Activision (USA)
Distribuzione italiana: Atari Italia (Milano)
Note: Ingresso solo step, e solo 4 tracce.
Prezzo (indicativo): Importato in Italia ≈ L. 99.000

Trasform-Xnotes™
Produttore: Beam Team (Usa)
Distribuzione italiana: Midi Ware (Roma)
Note: Composer che fa parte di un software integrato con diverse possibilità (Sequencer, Editor, Traduttore a Laserprint).
Compatibilità: Con gli altri programmi del set

Programmi CAI

Non ho notizia della loro esistenza però già da tempo si parla della traduzione (ri-compilazione) per Atari ST di diversi programmi tra i cui i più interessanti sono quelli della Musicom (Israele).
A livello orientativo i corsi Musicom (che in questo momento si trovano solo per hardware MS-DOS) comprendono:

Musicom™
Produttore: Xanadu International Ltd (Israele)
Distribuzione italiana: Music Technology (Milano)
Note:
A1-A2 Introduzione al canto e al riconoscimento delle note
B1 Elementi fondamentali della tastiera
P-EV Valutatore della tecnica pianistica
C1 Elementi di teoria della musica

C2 Concetti per la scrittura a due parti
C3/C3a Elementi di armonia: struttura delle triade
H1/H2 Corso di armonia applicata al pianoforte
P1 Corso rapido di pianoforte
J1/J2 Corso di tastiera jazz
R1 Corso di tastiera rock
W Corso orientato agli strumenti a fiato
Hardware speciale: Interfaccia MIDI, tastiera MIDI e microfono.
Valutazione complessiva: Eccellenti.

Programmi per la MIDI

I programmi per la MIDI costituiscono la maggior parte dell'offerta di software oggi; le differenze tra le diverse offerte presenti nel mercato, quando queste raggiungono un livello soddisfacente, puntano soprattutto alle "facilities". Non esiste ancora alcun programma "ottimale": quello che fa bene uno, un altro lo fa in modo macchinoso e viceversa.

Sequencer

Twentyfour 2.1™
Produttore: Steinberg (Germania occ.)
Distribuzione italiana: Midi Ware (Roma)
Note: Uno dei programmi sequencer più noti in ambiente ST. Viene presentato con la metafora del registratore a 24 tracce, si presenta con diverse pagine, ha una pagina per il "recorder", un'altra per l'"edit-score" (notazionale), un'altra per il "grid-edit" (editore grafico a griglia) e le diverse pagine per l'edizione e la gestione dei file. Le funzioni che ha implementate sono quantizzazione, "rasposizione, logical-edit ecc.
Permette di lavorare con pattern, massimo127 per traccia. Il massimo di eventi controllabili è 150.000.
Compatibilità: con tutti i programmi Steinberg.
Hardware speciale: No
Prezzo (indicativo): Importato in Italia ≈ L. 495.000

Iconix™
Produttore: System Exlusive Ltd label Tigress Designs (UK)
Note: Uno dei programmi sequencer più belli e potenti in ambiente ST. La metafora del registratore (in questo caso a 64 tracce) non è già più determinante

ma agevole e indicativa. Si presenta con diverse pagine e anche in multi-tasking; ha una pagine per il recorder, per l'editor, per il song-mode. Le funzioni implementate sono quelle standard, quantizzazione, trasposizione ecc. Inoltre il Midisong Pointer (che non si trova nella maggior parte dei programmi normali). Permette di lavorare con pattern che vengono intesi come microstrutture in song-mode, numero di eventi proporzionale alla memoria dell'elaboratore
Compatibilità: Con gli altri programmi della Tigress Designs.
Hardware speciale: No
Prezzo (indicativo): 250 sterline

Altri:

Creator™
Produttore: C lab (Germania occ.)
Distribuzione italiana: MIDI Music (Torino)
Note: Sequencer professionale 64 tracce.
Prezzo (indicativo): In Italia ≈ L. 640.000

EZ-track™
Produttore: Hybrid Arts (USA)
Distribuzione italiana: Mack Music (Roma)
Note: Sequencer + Composer 20 tracce
Compatibilità: Con tutti i prodotti Hybrid Arts
Prezzo (indicativo): ≈ L. 100.000

Glasstracks™
Produttore: Sonus (USA)
Distribuzione italiana: Music Pool
Note: Amatoriale. Sequencer 8 tracce.
Prezzo (indicativo): 40 $

KCS (Keyboard Controlled Sequenzer)™
Produttore: DR. T's (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Sequencer professionale, correttisimo nella sua concezione (tutte le funzioni sono implementate e memoria proporzionale).
Compatibilità: Con il Copyst™ e il MRS™
Prezzo (indicativo): ≈ L. 270.000

Masterpiece™
Produttore: Sonus (USA)
Distribuzione italiana: Music Pool
Note: Sequencer professionale fino a 32 tracce.
Prezzo (indicativo): ≈ 475 $

MIDIplay™
Produttore: Electronic Music Publishing House (USA)
Distribuzione italiana: Atari Italia
Note: Amatoriale, non sfrutta la metafora del registratore (un punto a suo favore) e riesce a registrare 250.000 eventi su ST 512 e più di 400.000 su 1040.
Prezzo (indicativo): ≈ L. 99.000

MIDI-Soft Studio™
Produttore: Midisoft Corp - Passport designs (USA)
Distribuzione italiana: Music Pool
Note: Sequencer 32 tracce.
Prezzo (indicativo): ≈ L. 150.000

MRS (Music Recording Studio)™
Produttore: DR. T's (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Amatoriale; sequencer piccolo (solo 8 tracce).
Compatibilità: Con il Copyst™ e il KCS™
Prezzo (indicativo): ≈ L. 80.000

Sync Track™ e SMPTE track™
vedere "sincronizzatori":

Super Conductor™
Produttore: Mich Tron (USA)
Distribuzione taliana: no
Note:
Amatoriale; sequencer (solo 16 tracce).
Prezzo (indicativo): 80 $

The Deck™
Produttore: Tigress Designs (UK)
Distribuzione italiana: no

Note: Amatoriale; sequencer (solo 16 tracce).
Prezzo (indicativo): 100 sterline

Trasform-XTrack™
Produttore: Beam Team (Usa)
Distribuzione italiana: Midi Ware (Roma)
Note: Sequencer che fa parte di un software integrato con diverse possibilità (Composer, Editor, Traduttore a Laserprint)
Compatibilità: Con gli altri programmi del set.

Editori di Parametri / Librerie

Questa è un altra famiglia numerosa; una prima classificazione permetterebbe di vedere due tipi, quelli "universali" ovvero per diverse macchine (il che li rende difficili da usare per le particolarità dei sistemi exclusivi ecc.) e quelli "dedicati", che sono per una periferica determinata. I primi, gli universali, li vedremo nella sezione sugli "utility tools". I secondi hanno una doppia classificazione: per sintetizzatori, e per controllo di periferiche di produzione (mixer MIDI) e post-produzione (rack effetti controllati via MIDI)

Editori dedicati a sintetizzatori

Elka EK 44™ Gen Patch Editor
Produttore: Mauro Graziani
Distribuzione: Tecnomusica (Milano)
Note: È il più bell'editore per uno strumento FM che abbia mai visto. Ha "facilities" per l'edizione di suoni FM che permettono finalmente di "capire" quello che si sta facendo (cosa che in FM è assai difficile), il "dumping" delle librerie è in modo interattivo e molto flessibile. In generale sfrutta al massimo le possibilità grafiche e interattive degli Atari ST.
Hardware dedicato: Elka mod. EK44

Casio CZ Patch™
Produttore: DR T's & Caged Artist (USA)
Distribuzione italiana: Music Technology
Note: Patch Editor - Librerie, grafico e interattivo.
Hardware dedicato: Serie CZ Casio™
Prezzo (indicativo): ≈ L. 150.000

CZ Droid™
Produttore: Hybrid Arts (USA)
Distribuzione italiana: Mack Music (Roma)
Note: Patch Editor - Librerie
Hardware dedicato: Serie CZ Casio™
Prezzo (indicativo): ≈ L. 185.000

DX Droid™
Produttore: Hybrid Arts (USA)
Distribuzione italiana: Mack Music (Roma)
Note: Patch Editor - Librerie, con generazione automatica di timbri.
Hardware dedicato: Serie DX Yamaha™
Prezzo (indicativo): ≈ L. 350.000

DX Heaven™
Produttore: DR T's & Caged Artist (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Patch Editor - Librerie, grafico e interattivo.
Hardware dedicato: Serie DX7 Yamaha™
Prezzo (indicativo): ≈ L. 150.000

DXware™
Produttore: Geerdes, Berlino (Germania)
Distribuzione italiana: Grisby Music
Note: Patch Editor - Librerie. Con generazione automatica.
Hardware dedicato: Serie DX7, TX ecc. Yamaha™

DX Pro Creator™
Produttore: Steinberg (Germany)
Distribuzione italiana: Midi Ware
Note: Patch Editor - Librerie, grafico e interattivo.
Hardware dedicato: Serie DX7 Yamaha™

FB01 DX-100 Patch™
Produttore: DR T's & Caged Artist (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Patch Editor - Librerie, grafico e interattivo.
Hardware dedicato: Serie Yamaha™
Prezzo (indicativo): ≈ L. 150.000

Transform Xsyn™ (FB01 DX-100 Patch)
Produttore: Beam Team (USA)
Distribuzione italiana: Midi Ware
Note: Patch Editor - Librerie, grafico e interattivo, generalizzato. fa parte del software integrato Transform.
Hardware dedicato: Tutti gli Yamaha™ FM e Casio CZ

Kawai K3 Patch™
Produttore: DR T's & Caged Artist (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Patch Editor - Librerie, grafico e interattivo.
Hardware dedicato: Kawai K3
Prezzo (indicativo): ≈ L. 150.000

Korg DP-2000/3000 Patch™
Produttore: DR T's & Caged Artist (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Patch Editor - Librerie, grafico e interattivo.
Hardware dedicato: Korg 2000/3000
Prezzo (indicativo): ≈ L. 150.000

Matrix-6 Patch™
Produttore: DR T's & Caged Artist (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Patch Editor - Librerie, grafico e interattivo.
Hardware dedicato: Oberheim Matrix-6
Prezzo (indicativo): ≈ L. 150.000

Roland D-50™
Produttore: DR T's & Caged Artist (USA)
Distribuzione italiana: Music Technology
Note: Patch Editor - Librerie, grafico e interattivo.
Hardware dedicato: Roland D-50
Prezzo (indicativo): ≈ L. 150.000

TR 707 Dumpstor™
Produttore: Music Services (USA)
Distribuzione italiana: no
Note: Libreria pattern

Hardware dedicato: Roland Drum Machine 707
Prezzo (indicativo): 65 $

Editori per il controllo di periferiche MIDI di produzione e postproduzione

Lexicon PCM 70 Patch™
Produttore: DR T's & Caged Artist (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Patch Editor.
Hardware dedicato: Lexicon PCM 70 Rack de effeti MIDI
Prezzo (indicativo): ≈ L. 150.000

C Mix (Kit)™
Produttore: Jellinghaus (Germania)
Distribuzione italiana: Music Technology (Milano)
Note: Package di automazione mixer, adattabile a banchi non MIDI attraverso una trasformazione hardware. Ha sincronizzazione MIDI e SMPTE.
Hardware dedicato: Qualsiasi banco fino a 16 tracce.

ATM CM16/8™
Produttore: ATM (Germania)
Distribuzione italiana: no
Note: Package automazione banchi mixer da non MIDI a MIDI con trasformazione hardware.
Prezzo (indicativo): ≈ 5000 DM

Generatori di dati in modo algoritmico

Fingers™
Produttore: DR T'S Music Software (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Programma di generazione algoritmica a quattro parti, l'utente attraverso una serie di parametri controlla la generazione e variazioni di 4 serie di eventi MIDI.
Compatibilità: Legge e scrive sui sequencer KCS e MRS della DR T's

PVG (Programmable Variations Generator)™
Produttore: DR T'S Music Software (USA)
Distribuzione italiana: Music Technology (Milano)
Note: Programma che può generare variazioni parametriche su qualsiasi parte memorizzata precedentemente, variazioni di frequenza, dinamica, ritmo ecc., la selezione delle note "da modificare" si effettua mediante operazioni di editing parziale e parametrico. Ad esempio la dinamica può essere aumentata solo per le note del primo quarto di ogni battuta oppure per le note più vicine a ogni semiminima pari o per le note che sono in una determinata posizione nella scala cromatica. Possono essere aggiunti valori di controllo che variano al variare della frequenza o la dinamica. Le note selezionate si possono raddoppiare su un altro canale oppure possono essere ricanalizzate, o cambiate in funzione degli accordi. Inoltre si può aggiungere "swing", ovvero variare leggeramente il ritmo in una battuta e anche variare il tipo di swing applicato da una battuta a un'altra.
Compatibilità: Legge e scrive dai sequencer KCS e MRS della DR T's.

Sincronizzatori

SMPTE track™
Produttore: Hybrid Arts (USA)
Distribuzione italiana: Mack Music (Roma)
Note: Sequencer 60 tracce con sincronizzatore hardware SMPTE to MIDI and back.
Compatibilità: Con tutti i prodotti Hybrid Arts
Hardware speciale: Interfaccia "box smpte" (inclusa nel prezzo).
Prezzo (indicativo): ≈ L. 1.000.000

SYNC track™
Produttore: Hybrid Arts (USA)
Distribuzione italiana: Mack Music (Roma)
Note: Solo la periferica hardware per SMPTE to MIDI and back.
Compatibilità: Con tutti i prodotti Hybrid Arts.
Prezzo (indicativo): ≈ L. 600.000

Utility tools

I programmi di utilità possono sono strumenti software di aiuto per lavori specializzati.

Utility per la stampa

Transform Xscript™
Produttore: Beam Team (USA)
Distribuzione italiana: Midiware
Note: Programma di utilità per stampare con stampante laser le partiture o le informazioni cretae dal pacchetto integrato Transform.
Compatibilità: Con tutti i prodotti del package.

MU- Script™
Produttore: Quiet Lion (USA)
Distribuzione italiana: no
Note: Programma di utilità per stampare partiture orchestrali
Compatibilità: Con tutti i prodotti della Quiet.

Utility per la MIDI

Data Dumpstore ST™
Produttore: Music Service Software (USA)
Distribuzione italiana: no
Note: Programma di utilità, generico per strumenti MIDI, caricatore universale di patch (per tutti i sintetizzatori)
Compatibilità: Con tutti.
Prezzo (indicativo): ≈ 90 $

Gen Patch™
Produttore: Hybrid Arts (USA)
Distribuzione italiana: Mack
Note: Programma di utilità caricatore universale di patch (per tuti i sintetizzatori).
Compatibilità: Con tutti.
Prezzo (indicativo): ≈ L. 265.000

MIDI Processor™
Produttore: Emmesoft (Italia)
Distribuzione italiana: Emmesoft (Torino)
Note: Programma di utilità generico per la MIDI.
Compatibilità: Con tutti gli strumenti MIDI.

MIDI Utilities™
Produttore: Hybrid Arts (USA)
Distribuzione italiana: Mack
Note: Programma di utilità che visualizza i dati MIDI in entrata e in uscita, può generare funzioni "macro", il che permette di realizzare qualsiasi tipo di lavoro con la MIDI; è anche logicamente un patch universale.
Compatibilità: Con tutti gli strumenti MIDI.

Formula™
Produttore: Anderson & Kuivila (USA)
Distribuzione italiana: Tecnomusica
Note: Kit di sviluppo software in linguaggio Forth, dedicato alla MIDI.
Prezzo (indicativo): Gratis, distribuito da Tecnomusica.

Programmi per l'elaborazione numerica del suono

Editori di campioni / Librerie

Akai S900 ™
Produttore: Geerdes (Germania)
Distribuzione italiana: Grisby Music
Note: Editore di parametri e di campioni.
Hardware speciale: S-900 Akai

Steinberg Sound Works Akai S900 ™
Produttore: Steinberg (Germania)
Distribuzione italiana: Midi Ware
Note: Editori di parametri e di campioni.
Hardware speciale: S-900 Akai
Prezzo (indicativo): ≈ L. 470.000

Steinberg Sound Works Mirage ™
Produttore: Steinberg (Germania)
Distribuzione italiana: Midi Ware
Note: Editori di parametri e di campioni
Hardware speciale: Campionatore Mirage Ensoniq

VDS (Visual Display System)™
Produttore: DR T's & Caged Artist (USA)
Distribuzione italiana: Music Technology
Note: Editore di parametri e di campioni, grafico e interattivo.
Hardware dedicato: Campionatore Ensoniq Mirage

Generatori di suoni per modelli di sintesi

Sampler Maker ™
Produttore: Bill Rosenkranz, Virtual Sounds (USA)
Note: Bellisimo e interessantissimo pacchetto, permette l'edizione di parametri e di campioni per Akai S900, Sequential Prophet 2000/2002, Ensoniq Mirage. Generazione di campioni via software per queste stesse macchine con due diversi modelli di sintesi: FM e Fourier (Sintesi additiva).
Prezzo (indicativo): ≈ L. 500.000

X Alyser ™
Produttore: C lab (Germania)
Distribuzione italiana: Midi Music (Torino)
Note: Trasforma i parametri DX in campioni numerici.
Prezzo (indicativo): ≈ L. 430.000

Hardware abbinati

Adap ™
Produttore: Hybrid Arts (USA7
Distribuzione italiana: Mack
Note: Campionatore hardware & software molto evoluto e di alta qualità controllato da ST, a 16 bit.
Compatibilità: Con qualunque campionatore (Emu, Akai ecc) .
Hardware speciale: Sound Rack

Prezzo (indicativo): ≈ L. 3.900.000

Soft Machine ™
Produttore: Steinberg (Germany)
Distribuzione italiana: Midi Ware
Note: Campionatore hardware & software, e drum machine controllato da ST a 16 bit.

ST - Replay ™
Produttore: Mich Tron (USA)
Distribuzione italiana: no
Note: Campionatore hardware & software non MIDI controllato da ST.
Prezzo (indicativo): ≈ 200 $

Hippo Sound Digitizer™
Produttore: Hippopotamus El (UK)
Distribuzione italiana: Atari Italia
Note: Campionatore hardware & software controllato da ST a 8 bit.
Hardware speciale: Scheda.
Prezzo (indicativo): ≈ L. 315.000

ST Sound Digitizer™
Produttore: Navarone Industries (USA)
Note: Campionatore hardware & software controllato da ST a 8 bit.
Hardware speciale: Scheda.
Prezzo (indicativo): ≈ 140 $

Sound / Streamer™
Produttore: York University (UK)
Distribuzione mondiale: Audio Design (UK)
Distribuzione italiana: no
Note: Trasferisce i sound file prodotti dagli elaboratori ST verso registratori digitali PCM 501/601/701.

Appendice 2

Who's Who

Quello che segue è un elenco di tutte le aziende e le organizzazioni, citate nel testo o nelle appendici, che sono in un modo o nell'altro attive nei settori di cui abbiamo parlato in queste pagine.

USA

Computer Music

International MIDI Association, 11857 Hartsook Street, North Hollywood CA 91607

Hardware

Atari Corporation, P.O. Box 61657, Sunnyvale CA 94088, ph (408) 745-2000

Periodici

Computer Music Journal, MIT Press Journals Dpt, 55 Hayward St, Cambridge MA 02142

Electronic Musician, Mix Publications, 2608 9th St, Berkley CA 94710

Keyboard, GPI Publications, P.O. Box 2110, Cupertino CA 95015

Keyboards Computers and Software, 299 Main St, Northport NY 11768

Music Technology, Music Maker Publications Inc, 7361 Topanga Canyon Blvd., Canoga Park CA 91303

STart, Antic Publishing, 524 Second Street, San Francisco CA 94107

Software musicale

Activision, 2350 Bayshore Parkway, Mountain View CA 94039 (USA), ph (415) 960 0410

Audio Light, 146 Town Terrace, Los Gatos CA 95030 (USA), ph (415) 344 4610

Beam Team, 6100 Adeline St., Oakland CA 94608 (USA), ph (415) 658 3208

Caged Artist Productions, 64 Griggs Rd , Brookline MA 02146, ph (617) 731 1948

DR T'S Music Software , 220 Bylston Street Suite 306, Chestnut Hill MA 02167 (USA), (617) 244 6954

Electronic Music Publishing House, 2210 Wilshire Blvd SUite 488, Santa Monica CA 90403, ph (213) 455 2025

Hybrid Arts, 11920 W Olympic Blvd, Los Angeles CA 90064, ph (213) 826 3777

XLENT, P.O. Box 5228, Springfield , VA 22150, ph (703) 644 8881

Key Clique, 3960 Laurel Canyon Blvd. Suite 374, Studio City CA 91604, ph (801) 566 1683

Mich Tron, 576 S Telegraph, Pontiac MI 48053

Micro W Distributing, 1342B Route 23, Butler, NJ 07405, ph (201) 838 9027

MIDI soft, P.O. Box 1000, Bellevue , WA 98009, ph (206) 827 0750

Navarone Industries Inc, 21109 Longeway Rd. Suite C, Sonora CA 95370

Passport Designs, 625 Miramontes St, Half Moon Bay CA 94019, ph (415) 726 0280

Quiet Lion, P.O. Box 219, Sun Valley CA 95370

Sonus, 21430 Strathern St Suite H, Canoga Parck CA 91304, ph (818) 702 0992

Virtual Sounds, P.O. Box 3286, Plimouth MA 02361

EUROPA

U.K.

Kuma Computers, 76w Horseshoe Park , Tangbourne Berkshire RG8

System Exlusive Ltd, Tigress Designs, 25 Burmesters Rd, London SW17

GERMANIA

C-lab, Postfach 710446, 2000 Hamburg 71

GC Geerdes, Guerichestrasse 43, D-1000 Berlin

JMS - Jellinghaus Music System, Martener Hellweg 40, 4600 Dortmund

Steinberg Research TSI, Neustrasse 12, 5481 Wadorf

ITALIA

AIMI (Associazione Informatica Musicale Italiana) C/O ASAC, Cà Corner della Regina S. Croce 2214, C/C Postale 10752301,30125 Venezia

Atari Italia SpA, Via dei Lavoratori 25, 20092 Cinisello Balsamo (Mi), tel (02) 612 08 51

Emmesoft, via Accademia Albertina 29, 10123 Torino

Grisby Music, via delle Industrie 4, Castelfidardo, tel (071) 781 714

Mack , via Tiburtina 364, 00159 Roma, tel (06) 431 704

Midi Music, via Cherubini 22, 10154 Torino, tel (011) 237 612

Midi Ware, via Parioli 101/C, 00197 Roma, tel. (06) 804 990

Music Pool, via Archimede 22, 20124 Milano, tel (02) 738 65 17

Music Technology, Res. Querce 311 - Milano 3, 20080 Basiglio (Mi), tel (02) 90 75 230

Tecnomusica, via Pastrengo 12A, 20159 Milano, tel (02) 683 221

ISRAELE

Xanadu International Ltd, P.O. Box 4503 , 91044 Jerusalem, ph (02) 63 49 73 (Israel)

Piccolo vocabolario dei termini informatici usati per l'Atari ST e nell'informatica musicale

I dizionari normalmente sono strettamente organizzati seguendo l'ordine alfabetico, questo invece funziona un po' come le pagine gialle: per voci.

Le unità di misura

Baud
Unità di misura standard della velocità di trasmissione dei dati calcolata in bit per secondo. Se leggi di una trasmissione con un *Baud rate* di 300 Baud significa che passano 300 bit al secondo dalla fonte alla destinazione.

Bit
È la più piccola unità di misura di memoria dei computer. Otto bit sono uguali a un byte

Byte
La memoria dei computer si misura in byte. Ogni byte consiste di otto bit.

High-Resolution (*Alta risoluzione*)
640 punti orizzontali per 400 punti verticali come risoluzione dello schermo (la risoluzione rende possibile una migliore definizione delle immagini). Gli Atari ST hanno tre possibilità di risoluzione. La *high* è solo monocromatica (bianco e nero); le altre due sono a colori.

IPS
Istruzioni per secondo, misura della velocità di proccesso della CPU.

Kilobyte (K)
1024 byte.

Low-Resolution (*Bassa risoluzione*)
Consiste in 320 punti orizzontali e 200 punti verticali come risoluzione dello schermo. In questa modalità gli Atari ST offrono anche la possibilità di 16 colori contemporaneamente, scelti da una tavolozza o *palette* di 512 colori.

Medium-Resolution (*Media risoluzione*)
Consiste in 640 punti orizzontali e 200 punti verticali, come risoluzione dello schermo. In questa modalità gli Atari ST offrono anche la possibilità di 4 colori contemporaneamente, scelti da una tavolozza o *palette* di 512 colori.

MIPS
Milioni di istruzioni al secondo (vedi IPS).

Pixel (*picture elements, componenti dell'immagine*)
L'immagine sullo schermo dei computer si costruisce a partire da punti; questi punti molto piccoli che costituiscono il minimo elemento di una immagine si chiamano "pixel". Lo schermo è una griglia di pixel e la definizione delle immagini aumenta quanti più pixel ci sono in un centimetro quadrato.

I tipi di azioni che puoi fare con il computer e i comandi più noti legati ad esse

(Questa è la lista meno esauriente di tutto il dizionarietto, però in questa lista troverai una grande quantità di chiarificazioni che difficilmente incontrerai nei manuali.)

Backup (*Riserva o scorta*)
Azione di creare una scorta, in informatica significa salvare da un posto ad un altro, normalmente indica il salvataggio su dischetto delle informazioni contenute in un hard disk: "Fare il backup del disco rigido". Si dice, più genericamente, anche del fare una copia di scorta: "ho fatto il backup del programma...".

Boot *(In gergo americano significa "lanciare", avviare)*
Significa avviare il sistema. Sul pannello posteriore degli Atari ST c'è un pulsante chiamato "Reset": quando lo si preme, il computer si riavvia, e questa azione si chiama "reboot". Per estensione il dischetto di avvio si chiama Boot disk.
Si definisce *boot* l'avviare l'installazione di qualche periferica, per esempio: "ho fatto il boot del disco rigido".

Cancel (*Annulla*)
È l'azione di annullare l'ultimo comando dato al computer; è una opzione in tutti i dialoghi; significa tornare alla situazione precedente.

Click (*Parola onomatopeica, in italiano si scrive in genere clic*)
Significa l'azione di premere il pulsante del mouse.

Close (*chiudere*)
Azione di chiusura di un documento o di una cartella; normalmente il computer ti chiederà attraverso una finestra di dialogo se vuoi salvare o no le variazioni fatte sul documento prima di chiuderlo.

Change (*Cambiare*)
Non è una azione primaria e normalmente la troverai in qualche applicazione tipo "word processor" o "data base", serve per cambiare una o tante volte quante vuoi una parola in un'altra.

Copy (*Copia*)
Azione di copiare. può effettuarsi a diversi livelli: puoi copiare parte di un documento, un documento completo, una cartella (o un gruppo di documenti), un disco ecc. L'azione è sempre la stessa, però a livello informatico sono diverse le modalità esecutive a seconda del livello. Copieari parte di un documento con l'uso del menu di *edit* e il comando *copy*, se vuoi fare copia di un documento nello stesso disco userai i comandi di duplicazione, invece se vuoi copiare documenti da un disco a un altro lo farai con il meccanismo *File copy* e se vuoi copiare il contenuto di un disco su un altro disco lo farai con il programma *Disk copy*.

Cut (*Taglia*)
Azione di tagliare una parte di un documento; il pezzo tagliato a partire da quel momento risiede in una memoria ausiliaria. Se con questo meccanismo si vuole trasportare parte di un documento in un altro documento e incollarlo lì, il processo di editing completo sarebbe: primo individuare il pezzetino in un documento, secondo tagliarlo con il *cut* (ovvero metterlo in memoria per trasferirlo), terzo chiudere il documento, quarto aprire il nuovo documento dove sarà incollato, quinto selezionare il punto in cui incollarlo, sesto *paste*, ovvero incollarlo.
Normalmente questo comando si trova nel menu "edit". Il comando *copy* in questi tipi di menu permette lo stesso risultato, però senza tagliare dal primo documento il pezzo che si vuol trasferire.

Delete (*Distruggere, Cancellare*)
Normalmente é il comando più pericoloso perche dopo averlo usato non cè niente da fare: tutto perso per sempre. Negli Atari ST, per cancellare, si usando la "pattumiera" o *trash*, in cui si portano e si buttano i documenti. Attenzione, però!

Diskcopy (*Copia disco*)
(Vedere *copy*) È un programma che serve per copiare dischi. Con la prattica potrai verificare che i programmi che copiano si chiamano "copiatori", e anche che la maggior parte dei programmi che girano sono "protetti", ovvero che non possono essere copiati con programmi di questo tipo. Comunque vedrai anche girare qualcuno di questi programmi sprotetto, perche qualcuno è riuscito a scoprire ed eliminare la protezione. Diffida assolutamente delle copie sprotette, soprattutto se te le vogliono fare pagare a un qualsiasi prezzo. I programmi sono gli strumenti del tuo lavoro e il loro prezzo (salvo eccezioni)

risponde alla quantità di lavoro svolta dai loro creatori e anche al risparmio di tempo/lavoro che ti procureranno. I programmi (fatta eccezione per quelli del *free-software* o per quelli di dominio pubblico) richiedono una "manutenzione", che si concretizza anche nella commercializzazione di versioni successive; per poter essere usati correttamente vengono venduti insieme a una guida d'uso o *User's Manual* e in genere anche a una documentazione più tecnica, il *Reference manual* ed educazionale, il *Tutorial manual.*
L'uso di software copiato non è un problema esenzialmente economico, ma piuttosto un problema culturale: il costo della fotocopia dei manuali, che può essere superiore al costo del pacchetto (se riesci a trovarli) e l'incertezza sull'affidabilità del software secondo me sono due argomenti da soppesare. Con i programmi originali troverai una scheda da riempire e spedire alla software house originaria o al suo rappresentante nell'area territoriale corrispondente, e questa scheda costituisce una "garanzia" di uso e manutenzione, e allo stesso tempo ti permette un contatto diretto con il creatore del prodotto.

Do (*Fai!*)
Imperativo di: fare!

Double-click (*Doppio-clic*)
Per aprire un'icona è neccesario fare un doppio-clic col mouse. L'intervallo di tempo massimo fra i due clic perché vengano accettati come un solo comando (e non come due clic separati) può essere regolato dall'utente attraverso il "control panel".

Dragging (*Trascinare*)
È la tecnica per muovere le informazioni nell'ambiente di lavoro del computer (metaforicamente una scrivania = *desktop*). Usando il mouse "trascinerai" le immagini degli oggetti (*items*) che vuoi portare da un posto ad un altro e il computer eseguirà effettivamente l'operazione descritta sullo schermo. Per esempio, per portare un documento da un dischetto ad un altro, trascinerai l'icona che rappresenta il documento che vuoi trasferire dalla finestra del dischetto A alla finestra del dischetto B, dove lo rilascerai; lì comparirà di nuovo l'icona, e da quel momento il computer eseguirà la tua decisione transferendo sul dischetto B le informazioni che erano sul dischetto A.

Edit (*Edizione*)
È uno dei menu più utili e popolari delle macchine con filosofia *end user*, relativo alle attività di modifica e correzione. Mostra una serie di processi

concatenabili, che sono generalmente *copy*, *erase*, *cut* e *paste* (vedi i singoli termini).

Erase (*Cancellare - con la gomma*)
In omaggio alla metafora della scrivania, significa proprio quello: cancellare qualcosa. Il meccanismo non permette il recupero di quello che è stato cancellato.

Escape (*Uscita*)
È il comando (il tasto) che ti permette di uscire, da un programma, da un documento ecc. Si trova nell'angolo sinistro in alto della tastiera (*Esc*)

Find (*Trovare, scoprire*)
È uno dei comandi dei menu di ricerca (Search). Avvia un programma che si occupa di "trovare", a partire da una informazione, il punto (o i punti) di un documento in cui quell'informazione si trova.

Format (*Formato*)
L' informazione viene inmagazzinata su un dischetto in porzioni circolari. Quando si ***formatta*** un dischetto si creano gli spazi in cui verrà immagazzinata l'informazione; il comando dunque crea una organizzazione del dischetto mettendolo in condizioni di ricevere le informazioni, però conviene stare attenti con questo comando perché quando viene eseguito tutto quello c'era prima sul dischetto viene irreparabilmente distrutto.
Ogni elaboratore formatta i dischi in maniera diversa.
Gli Atari ST "leggono" bene i dischetti prodotti per elaboratori MS-DOS (ed anche per il nuovo System 2 della IBM) ed elaboratori MSX, ma non vale il viceversa perché la formattazione è diversa.
Nota Bene: A causa del gran numero di tipi diversi di dischetti da 3,5 pollici (soprattutto in ambito di informatica musicale) è bene stare molto attenti con il comando di formattazione, perché, per esempio, un dischetto di un DX7 o un dischetto di un camionatore, o anche di un qualsiasi altro calcolatore (fatta eccezione per quelli citati) vengono visti dall'Atari come dischetti "vergini", e subito l'elaboratore di chiede se vuoi formattarli. E se dai il comando, addio per sempre a tutto quello che c'era dentro (come si dice in inglese, "Bye forever")!

Go to (*Vai a*)
Istruzione elementare in molti linguaggi di programmazione: dice

all'elaboratore di andare in un certo punto, definito subito dopo l'istruzione.

Info (*Informa*)
È un comando per chiedere informazioni. Si gestisce selezionando l'icona di ciò su cui si vogliono avere informazioni, quindi dando il comando *Show Info* (mostra informazioni).

Install (*Installa*)
Comando per l'installazione di una periferica o di un programma che richiede un'installazione "non standard".
• Install printer: installa la stampante (vedi Print).
• Install disk drive: installa il lettore di dischetti esterno.

Help (*Aiuto*)
La maggior parte delle applicazioni contiene questo comando, che attiva un programma, di carattere esplicativo, legato ai diversi punti in cui possono sorgere dubbi sulla gestione dell'applicazione: sullo schermo compare un testo che spiega che cosa si può (o si deve) fare in quel momento. Si attiva con un tasto funzionale da tastiera (tasto Help) o tramite una opzione da menu.

Load (*Leggi, carica*)
È il comando di lettura in memoria. Normalmente il processo è lo stesso di *Open*, che però è più specifico dei documenti.

Paste (*Incolla*)
Azione d'incollare: è un comando di menu Edit.

Print (*Stampa*)
Comando di stampa. Con questo comando si avvia un programma che permette la stampa di documenti con le periferiche apposite (DMP, stampante laser, plotter). Non tutte le stampanti sono uguali: esistono modelli diversi che rispondono in modo diverso, qualcuno compatibile con gli Atari e altri no. L'Atari è compatibile con lo standard Epson-80 per le DMP; per le stampanti laser, Atari ne ha una propria, collegabile ai modelli Mega. Ogni stampante richiede l'installazione software dei propri "driver" (così si chiamano i programmi che controllano queste perfieriche), e quindi il Print funziona solo se la stampante è compatibile, è "installata" ed è collegata correttamente. In caso di mancato funzionamento di una stampante al comando Print, conviene leggere attentamente le istruzioni.

• Print Screen: comando che permette di stampare tutto quello che si vede sullo schermo degli ST.

Quit (*Esci*)
Comando che permette di uscire da un programma e tornare allo schermo iniziale.

Retry (*Riprova*)
Comando di solito abbinato al comando Cancel, nelle finestre di dialogo: permette di riprovare un'operazione.

Save (*Salva*)
Salva in memoria o registra.

Save as... (*Salva come...*)
Con questo comando l'applicazione non registra il documento che è stato aperto, ma ne genera uno nuovo con tutte le informazioni del precedente, e permette di attribuirgli un nuovo nome.

Scroll (*Scorrimento*)
Quando i documenti sono di lunghezza superiore a una singola schermata ci sono due possibilità: vederli a pagine (ogni schermata una pagina) o farli scorrere sullo schermo. Questa seconda possibilità è lo "scroll".

Set (*To set = mettere, porre, stabilire, fissare*)
Azione di scelta o definizione di parametri.
• Set Screen Resolution: sceglie il tipo di risoluzione dello schermo
• Set Preferences: stabilisce le preferenze
• Set RS 232 Config: stabilisce la configurazione della porta RS 232.

Setup (*Messa a punto, condizioni di avviamento*)
Insieme dei parametri di avviamento.

Show (*Mostra*)
Azione di mostrare.
• Show Info: mostra informazioni
• Show as Icons: mostra gli oggetti come icone
• Show as Text: mostra gli oggetti come testo.

Sizing (*Dimensionamento*)
Comando che dà la dimensione di occupazione di memoria dei singoli oggetti informatici (file, cartelline, dischi ecc.). Conoscere in ogni momento l'occupazione di memoria è fondamentale per lavorare tranquilli, perché le macchine hanno una capacità massima, come i dischetti, ecc.

Sort (*Ordina*)
Comando che avvia un piccolo programma di ordinamento delle informazioni:
- by Name: in ordine alfabetico
- by Date: per data
- by Size: per ordine di grandezza
- by Type: per tipi e, a parità di tipo, in ordine alfabetico.

Undo (*Disfare*)
Comando simile a Cancel, ma a livello di applicazione: con questo comando viene "disfatta" l'ultima azione effettuata, riportando alla condizione precedente.

Gli oggetti informatici virtuali

Sono gli oggetti che vengono presentati sullo schermo, e che concretizzano la metafora dell'ambiente di lavoro come scrivania (*desktop*).

Alert messages (*Messaggi di allerta*)
Sono messaggi di avvertimento che compaiono nelle finestre di dialogo (dialog boxes); avvertono che l'operazione che si vuol compiere è potenzialmente pericolosa, impropria o impossibile.

Close Box (*Chiudi scatola*)
Nell'angolo superiore sinistro della finestra attiva (*active window*) c'è un piccolo quadratino con una specie di croce; posizionando su di esso il cursore e facendo clic si può chiudere la finestra.

Color Palette (*Tavolozza di colori*)
Funzione del pannello di controllo, che permette di modificare i colori degli elaboratori ST. Per poterla usare è necessario avere l'SC1224 RGB Analog Color Monitor, o un televisore a colori collegato all'ST, mentre con il monitor in bianco e nero questa funzione non serve a nulla...

Control Panel (*Pannello di controllo*)
Con l'omonimo comando del menu Desk si fa comparire sullo schermo una finestra di dialogo: è il pannello di controllo, dal quale si possono regolare varie funzioni del GEM desktop, come *Color Palette*, *Mouse Click Response* (la velocità di risposta del mouse), *Keyboard Response* (sensibilità di risposta dei tasti della tastiera alfanumerica), *Clock Calendar* (Orologio e calendario), *Audio/Feedback* (Segnali di avviso acustici).

Cursor (*Cursore*)
Sullo schermo appare sempre un trattino intermittente, che indica qual è la posizione in cui si scriverà il prossimo carattere: è il cursore.

Desk Accessory (*Accessorio di scrivania*)
Gli accessori di scrivania si trovano sotto il menu Desk, e possono essere tanti quanti si vuole. Come file hanno sempre il suffisso .ACC. Agli accessori si può accedere dallo schermo iniziale, la scrivania del GEM, o mentre si sta usando un qualsiasi programma. Gli accessori permettono di trasferire facilmente informazioni tra programmi diversi.

Dialog Box (*Finestra di dialogo*)
Sono le finestre rettangolari che compaiono al centro dello schermo e permettono l'interattività: possono fornire informazioni, o presentare opzioni diverse tra cui scegliere.

Full Box (*Scatola piena*)
La traduzione letterale del termine inglese può dare un'idea sbagliata. Corrisponde alla possibilità di aprire ogni finestra al massimo dello schermo (*full screen*); come nel caso del *Close Box*, si attiva posizionando la freccia (il puntatore) sull'angolo superiore sinistro della finestra attiva e facendo clic.

GEM desktop (*Scrivania del GEM*)
GEM (*Graphics Environment Manager*, gestore dell'ambiente grafico) è un programma usato nel TOS (il sistema operativo degli ST) per creare e gestire le finestre, le icone, i menu e tutte le possibilità grafiche degli elaboratori Atari ST e Mega. La scrivania del GEM corrisponde allo schermo principale (*main screen*) dove si effettuano tutte le operazioni di alto livello. Fondamentalmente contiene il menu bar o striscia dei menu, le icone di due floppy disk e l'icona della pattumiera (*trash*).

Ghost (*Fantasma, spettro, falsa immagine*)
Quando trasciniamo (*dragging*) delle icone da un punto all'altro sullo schermo, con la freccetta si muove un'immagine, che rappresenta il contorno degli oggetti che stiamo trascinando: questa immagine è il *ghost*.

Icon (*Icona*)
Nella metafora della scrivania, gli oggetti informatici (file, cartellette, programmi) vengono rappresentati da figure, cioè icone.

Information Line (*Riga di informazione*)
È la linea che si trova nella parte superiore della finestra attiva e fornisce le informazioni sul numero dei byte usati e sul numero degli elementi (*item*) presenti.

Item (*Elemento*)
Anche: numero di items = numero di elementi.

Menu Bar (*Striscia dei menu*)
Nella parte superiore dello schermo principale nella scrivania del GEM c'è una striscia bianca dove appaiono i nomi dei menu. Questa è la "menu bar", e i nomi dei menu iniziali sono Desk, File, View, Options; quando poi si usano programmi particolari, questi usano la menu bar per offrire i loro menu specifici.

Move Bar (*Striscia per muovere*)
È la striscia superiore delle finestre, piena di puntini, dove si trova anche l'indicazione del dischetto (A: o B:, ecc.). Posizionando su questa striscia la freccia del puntatore si può trascinare l'intera finestra (come si fa con le icone).

Pointer (*Freccetta, puntatore*)
È detto anche "mouse pointer".

Scroll Bar (*Striscia per scorrere*)
Nelle finestre attive, a destra o in basso si possono trovare delle strisce che portano ai loro estremi delle freccette: sono le scroll bar, abilitate quando la finestra contiene più informazione di quella visibile; hanno una parte ombreggiata e una parte in bianco, le cui dimensioni indicano le proporzioni fra la parte visibile (bianco) e la parte non visibile (ombreggiato). Si possono usare in vario modo.

• Scroll lento: facendo clic alle freccette sugli estremi, il contenuto scorre, stringa per stringa.
• Scroll a step: facendo clic sulla parte ombreggiata scorre un'intera pagina in quella direzione.
• Scroll veloce: trascinando la parte bianca e poi lasciandola, compare la videata corrispondente, proporzionalmente, alla nuova posizione.

Size Box (*Scatola di grandezza*)
Si trova nell'angolo inferiore sinistro della finestra attiva; permette di portare le finestre alla grandezza che si desidera. Si posiziona il puntatore su quel punto, si preme il mouse e lo si tiene premuto: compare il ghost della finestra e su quello ci si regola, muovendo il mouse, per stabilire la dimensione voluta. Quando si lascia il mouse la finestra risponde e si dimensiona come vogliamo.

Trash (*Pattumiera*)
Questa icona si usa sulla scrivania del GEM per cancellare file o cartellette. Quando un file o una cartelletta sono messi nella pattumiera se ne vanno per sempre!!

Window (*Finestra*)
È l'area dello schermo in cui appaiono le icone dei file e delle cartellette; nella metafora del GEM è l'area di lavoro, dove si possono leggere o duplicare file e cartellette, dove si possono avviare i programmi. La scrivania del GEM permette un massimo di quattro finestre aperte allo stesso tempo.

L'organizzazione delle informazioni

Nell'ambiente degli ST le informazioni si trovano sotto tre tipi fondamentali di icone, che corrispondono a:

Data File (*File di dati*)
Sono una serie di informazioni specifiche usate o prodotte da un programma. I Data File non sono programmi, e nella rappresentazione grafica del GEM desktop hanno un'icona caratteristica: una serie di fogli di carta, il primo dei quali con l'angolo inferiore destro piegato.

Folder (*Cartelletta*)
I file si possono raggruppare in cartellette; per aprire una cartelletta basta fare

un doppio clic sulla sua icona (che ha la forma di una tipica cartelletta sospesa da classificatore).

Program File (*File di programma*)
È un file che contiene un programma o applicazione. Le icone dei programmi sono diverse dalle altre icone. I programmi sono rappresentati da un rettangolo con una striscia ombreggiata nella parte superiore; dopo il nome portano il suffisso .PRG. Avviare o lanciare un programma dalla scrivania del GEM è semplice: basta posizionare il puntatore sull'icona relativa e fare un doppio clic veloce con il mouse.

Bibliografia

Per il capitolo 1

Arnold Schoenberg, *Manuale di Armonia*, Il Saggiatore Milano 1963

A. Robertson & D. Stevens, *The Pelican History of Music*, Penguin Books England.

T. Dubois, *Trattato di Contrappunto e Fuga*, Ricordi Milano 1947

Paul Hindemith, *Armonia Tradizionale*, Curci Milano 1955

Otto Karolyi, *La grammatica della Musica*, Einaudi 1970 PBE 119

R. Stephan, *Musica*, ed. it. a cura di L. Pestalozza, Feltrinelli, Milano 1967.

M. Ravel, *Bolero*, Durand S.A. Paris 1929

Roland de Candé, *Diccionari de la Musica*, edicions 62 Barcelona 1969

P. Boulez, *Penser la Musique aujourd'hui*, Paris 1972

Atari ST™ Owner's Manual, Atari corp. Sunnyvale CA 1986.

M. Maiocchi, *Teoria e Applicazioni delle macchine calcolatrici*, Casa Editrice Ambrosiana Milano 1984.

Per il capitolo 2

Midi 1.O Detailed Specification, by International Midi Association CA
Bob Moog, "*MIDI*", *J. Audio Eng. Soc.* Vol 34 n. 5, May 1986
Craig Anderton, *MIDI for Musicians*, Amsco Publications New York 1986
Cristopher Yavelow, "Personal Computer and Music the state of the art", *J. Audio Eng. Soc.* Vol 35 n. 3, March 1987
Fiorella Terenzi, "Midi Music", *Riv. Atari* Jackson Milano n. 4, Agosto1987
Dario Bontempi, *Interfaccia MIDI per PC*, Tesi di Laurea Politecnico di Milano corso 85-86.

Per il capitolo 3

Goffredo Haus, *Elementi d'Informatica Musicale*, Jackson Milano 1985
J. Moorer, Curtis Abbott ed altri, *The Digital Audio Processing Station*, J. Audio Eng. Soc. Vol 34 n. 6, June 1986
Gareth Loy, "Musicians Make a Standard: The Midi Phenomenon", *Computer Music Journal* vol 9 n. 4, Winter 1985.
A. R. Brinkman, *Representing Musical Scores for Computer Analysis*,
G. degli Antoni, G. Haus, *Music and Causality*, Computer Music Association Ed. San Francisco 1983.
C. A. Petri,*General Net Theory*, Proceedings of the Joint IBM & Newscastle upon Tyne Seminar on Computer Systems Design 1976.
S. Pope, "The Development of an Intelligent Composer's Assistant", *Proceedings of the 1986, International Computer Music Conference Den Haag*, San Francisco 1986
G. Haus A. Rodriguez, *Music Description and Processing by Petri Nets*, Agosto1987.
A. Camurri, G. Haus, R. Zaccaria, "Describing and Performing Musical Processes by Means of Petri Nets", *Interface* vol 16 n. 1, Amsterdam 1986;
A. Camurri, G. Haus, R. Zaccaria, "Music Knowledge Representation: Analogic and LogicSymbolic Systems", *Proceedings IV International Conference of Event Perception and Action*, Università di Trieste 1987.
A. Camurri, G. Haus, G Jacomini, R. Zaccaria, "Il sistema MAP per il controllo del CMI Fairlight", *Atti del 6° colloquio di Informatica Musicale*, Università di Napoli, UNICOPLI Milano 1987.

Per il capitolo 4

Alan R Feuer & Narain H Gehani (Bell Laboratories), "A comparison of programming languages C and Pascal", *Computing surveys* vol 14 n. 1, March 1982

N. Wirth, *The Programming Language Pascal*, Acta Informatica 1 1971

Jensen K. and Wirth N., *The Pascal user manual and report*, Springer Verlag New York 1974.

Kernighan B.W. and Ritchie D.M., *The C programming language,* Prentice Hall Engelwood Cliffs New York 1978

Giancarlo Mauri, *Comunicare con il computer: I linguaggi*, Jackson Milano1986.

I computer Atari della serie ST sono macchine personali estremamente flessibili e molto potenti, con una interfaccia "amichevole" verso l'utente, che nascono già con una precisa vocazione musicale: sono gli unici personal che abbiano incorporata l'interfaccia MIDI (Musical Instruments Digital Interface). *Non sono macchine per giocare: il musicista, dal dilettante al professionista, può trovarvi motivo d'interesse e di impiego immediato. Questo volume vuole proprio essere una guida all'uso più proficuo dei computer Atari sotto il profilo musicale: non presuppone alcuna conoscenza, né di informatica né di musica, ma conduce il lettore dalle nozioni di base fino alle applicazioni più avanzate, come la gestione della MIDI, l'uso del computer per la composizione, l'orchestrazione, l'analisi musicologica, l'editoria musicale. Le appendici forniscono una panoramica aggiornata del software musicale disponibile commercialmente per gli Atari ST, una rassegna di tutte le società e gli enti che lavorano nel campo dell'informatica musicale su queste macchine, un glossario di informatica musicale.*

Toni Rodriguez, pianista e compositore, ha seguito studi musicali e scientifici nella sua città natale, Valencia in Spagna. Fra il 1969 e il 1975 si è dedicato contemporaneamente alla ricerca musicologica sulla musica folkloristica spagnola e a sperimentazioni di musica elettronica. Dal 1975 al 1983 è attivo in tutta Europa come esecutore, e come direttore di produzione discografica. Specializzatosi in ingegneria del suono e in computer music, dal 1985 lavora a Milano con Goffredo Haus a un progetto di ricerca sugli algoritmi della creazione musicale.

strumenti della musica

L. 24.000 (23.529) ISBN 88-7021-427-3

23

Fare musica con Atari

TONI RODRIGUEZ